OPTIMASI PRODUKSI MENGGUNAKAN METODE *FUZZY LINEAR PROGRAMMING*
(Studi Kasus di *Home* Industri ‘Amanah’ Kediri)

SKRIPSI

Oleh:
HANDYGA PUTRA MUSPA ASTONIS
0810953041-95

PROGRAM STUDI STATISTIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2014

OPTIMASI PRODUKSI MENGGUNAKAN METODE *FUZZY LINEAR PROGRAMMING*
(Studi Kasus di *Home* Industri 'Amanah' Kediri)

SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains dalam bidang Statistika

Oleh:
HANDYGA PUTRA MUSPA ASTONIS
0810953041-95

PROGRAM STUDI STATISTIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2014

LEMBAR PENGESAHAN SKRIPSI

OPTIMASI PRODUKSI MENGGUNAKAN
METODE *FUZZY LINEAR PROGRAMMING*
(Studi Kasus di *Home* Industri 'Amanah' Kediri)

Oleh :
HANDYGA PUTRA MUSPA ASTONIS
0810953041

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 21 Juli 2014
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam Bidang Statistika

Dosen Pembimbing I	Dosen Pembimbing II
Eni Sumarminingsih, S.Si., MM.	**Samingun Handoyo, S.Si., M.Cs.**
NIP. 197705152002122009	**NIP. 197304151998021002**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc.
NIP. 19670907 199203 1 001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Handyga Putra Muspa Astonis
NIM : 0810953041
Jurusan : Matematika
Penulis Skripsi Berjudul : Optimasi Produksi Menggunakan Metode *Fuzzy Linear Programming* (Studi Kasus di *Home* Industri 'Amanah' Kediri)

Dengan ini menyatakan bahwa :

1. **Skripsi ini adalah benar-benar karya saya sendiri dan bukan hasil plagiat dari karya orang lain. Karya-karya yang tercantum dalam Daftar Pustaka Skripsi ini, semata-mata digunakan sebagai acuan/referensi.**
2. **Apabila dikemudian hari diketahui bahwa isi Skripsi saya merupakan hasil plagiat, maka saya bersedia menanggung resiko yang akan saya terima.**

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 21 Juli 2014
Yang menyatakan,

Handyga Putra Muspa Astonis
NIM. 0810953041

OPTIMASI PRODUKSI MENGGUNAKAN METODE *FUZZY LINEAR PROGRAMMING* (Studi Kasus di *Home* Industri 'Amanah' Kediri)

ABSTRAK

Permasalahan biaya produksi dan pengendalian bahan baku merupakan bagian penting dalam sektor produksi yang perlu dioptimalkan. Prosesnya adalah mengkoordinasikan produksi dan stok, sehingga diperoleh solusi optimal atau *near-optimal* (Susanto dan Sarwadi, 2006). Mengingat berbagai kemungkinan buruk bisa saja terjadi selama proses produksi, maka sangat perlu bagi *home* industri 'Amanah' untuk melakukan optimasi produksi dengan metode *Fuzzy Linier Programming*. Tujuan dari penelitian ini adalah menggunakan metode *Linear Programming* dan *Fuzzy Linear Programming* untuk menemukan optimasi produksi *home* industri 'Amanah'. Serta mermbandingkan hasil optimasi antara *Linear Programing* dengan *Fuzzy Linear Programming*. Hasil dari perhitungan *Linear Programming* biasa, keuntungan maksimum yang dapat diterima oleh *home* industri 'Amanah' dalam memproduksi abon ayam sebesar Rp. 57.500,-/hari dengan harus memproduksi abon ayam krispi manis sebanyak 2 kali produksi dan abon ayam krispi pedas sebanyak 2 kali produksi dalam sehari. Hasil dari perhitungan *Fuzzy Linear Programming*, keuntungan maksimum yang dapat diterima oleh *home* industri 'Amanah' dalam memproduksi abon ayam adalah sebesar Rp. 60.375,-/hari (Rp. 2.875,- lebih banyak dibanding dengan *Linear Programming* biasa) dengan harus memproduksi abon ayam krispi manis sebanyak 2 kali produksi dan abon ayam krispi pedas sebanyak 3 kali produksi dalam sehari. Artinya, penyelesaian *Fuzzy Linear Programming* akan memberikan hasil lebih baik dan optimal jika dibandingkan dengan penyelesaian *Linear Programming* biasa.

Kata kunci : Optimasi, *Linear Programming*, *Fuzzy Linear Programming*

PRODUCTION OPTIMIZATION USING FUZZY LINEAR PROGRAMMING METHODS
(Case Studies in Home Industries ‘Amanah’ Kediri)

ABSTRACT

Production costs and raw materials controlling problems is an important part in the production sector, which needs to be optimized. The process is coordinating production and stocks, in order to obtain the optimal solution or near-optimal (Susanto and Sarwadi, 2006). Given the variety of possible bad could happen during the production process, it is necessary for the home industry 'Amanah' to perform production optimization with Fuzzy Linear Programming. The purpose of this research is to use Linear Programming and Fuzzy Linear Programming methods to find the optimization of production home industry 'Amanah'. Comparing and optimization results between Linear Programming with Fuzzy Linear Programming. The results of the calculation of Linear Programming, the maximum benefit that can be received by home industry 'Amanah' in producing chicken shredded is Rp. 57.500, - / day with sweet crispy chicken shredded should produce as much as 2 times the production and spicy crispy chicken shredded production 2 times a day. The results of the calculation of Fuzzy Linear Programming, the maximum benefit that can be received by home industry 'Amanah' in producing chicken shredded is Rp. 60 375, - / day (Rp. 2,875, - much more than the Linear Programming) with sweet crispy chicken shredded should produce as much as 2 times the production and spicy crispy chicken shredded production 3 times a day. That is, the completion of Fuzzy Linear Programming will give better results and optimal when compared with the normal completion of Linear Programming.

Key words : Optimization, Linear Programming, Fuzzy Linear Programming

KATA PENGANTAR

Alhamdulillahi Robbil ’alamiin, puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi yang berjudul ” *Optimasi Produksi Menggunakan Metode Fuzzy Linear Programming (Studi Kasus di Home Industri ‘Amanah’ Kediri)*”. Sholawat serta salam semoga terlimpahkan kepada Rasulullah Muhammad SAW. Dengan terselesaikan penyusunan skripsi ini, penulis mengucapkan terima kasih kepada :

1. Ibu Eni Sumarminingsih, S.Si., MM., selaku dosen pembimbing I dan dosen Pembimbing Akademik yang telah memberikan motivasi, bimbingan, masukan, dan pengarahan hingga penulis menyelesaikan skripsi dan kuliah dengan baik.
2. Bapak Samingun Handoyo, S.Si., M.Cs., selaku dosen pembimbing II yang telah memberikan motivasi, bimbingan, masukan dan pengarahan hingga skripsi ini terselesaikan dengan baik.
3. Ibu Ir. Heni Kusdarwati, MS., selaku dosen penguji atas saran dan masukan yang telah diberikan.
4. Bapak Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika FMIPA Universitas Brawijaya Malang.
5. Staf pengajar Statistika dan administrasi Jurusan Matematika FMIPA Universitas Brawijaya Malang atas ilmu pengetahuan, semangat dan bantuan yang diberikan.
6. Bapak Musiran, Ibu Partinah, Mas Happy Putra, Chainis, dan semua keluarga atas dukungan dalam setiap jalan yang ditempuh penulis.
7. Haris, Vier, Angga, Izul, Ajeng, Dwi Masrokah, Ani, Mufira, Rifal, Myta, Fahmi, Fitri, Andika, Sakib, dan Deni yang setia menemani untuk menyelesaikan penulisan skripsi ini.
8. Teman-teman statistika dari berbagai angkatan *espesially* Statistika 2008 ”*One For All*, *All For One*” atas perhatian, perjuangan, dukungan, kerjasama dan semangatnya selama ini.
9. Penghuni jl. Vinolia gang VII no. 26A yang telah memberikan motivasi dan kekeluargaan selama menjalani kuliah di sini.
10. Seluruh pihak yang telah berpartisipasi yang tidak dapat penulis sebutkan seluruhnya.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis menerima saran dan kritik yang membangun demi kesempurnaan skripsi ini. Semoga skripsi ini ini dapat bermanfaat bagi banyak pihak.

Malang, Juli 2014

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1. Latar Belakang

Bagi perusahaan yang melakukan kegiatan produksi, persediaan bahan baku merupakan faktor paling utama, karena tanpa persediaan bahan baku yang cukup, produksi akan terhambat. Besar kecilnya persediaan yang dimiliki sangat tergantung pada kebijakan perusahaan, dan hal ini ditentukan dengan pertimbangan tertentu salah satunya adalah faktor biaya (Fisiana, 2010).

Permasalahan biaya produksi dan pengendalian bahan baku merupakan bagian penting dalam sektor produksi yang perlu dioptimalkan. Prosesnya adalah mengkoordinasikan produksi dan stok, sehingga diperoleh solusi optimal atau *near-optimal*. Karena itu, analisa dan optimasi biaya produksi harus dilakukan oleh manajemen perusahaan. Sebab, keberhasilan optimasi biaya produksi akan memberikan penghematan yang bisa dilokasikan pada devisi lain (Susanto dan Sarwadi, 2006).

Secara praktis, optimasi harus dilakukan secara berurutan dan mengikuti standar penanganan yang telah dibakukan perusahaan. Beberapa proses diantaranya melibatkan mesin dan manusia yang memiliki ketidakpastian waktu, kualitas hasil dan jumlah. Dengan demikian, tidak mudah untuk menentukan dan menemukan proses mana yang berjalan tidak optimal, sebab skala produksi yang besar tentu memiliki jumlah proses yang tidak sedikit dan membutuhkan waktu dan biaya yang besar dalam pengoperasiannya. Akibatnya, jika dilakukan pengujian secara riil untuk melakukan optimasi akan membutuhkan waktu lama dan membebani perusahaan dengan biaya yang besar (Fisiana, 2010).

Home industri ‘Amanah’ merupakan industri yang bergerak di bidang makanan jadi, yang memproduksi dua jenis abon ayam (krispi manis dan krispi pedas). Kebijakan pemilik usaha untuk menetapkan harga, berdasarkan biaya produksi seperti biaya bahan baku dan tenaga kerja. Tiap produksi *home* industri ‘Amanah’ mampu menerima laba 300 ribu rupiah untuk krispi manis dan 450 ribu rupiah untuk krispi pedas. Hanya saja, selama proses produksi dilakukan belum pernah sekalipun dilakukan evaluasi aspek mana yang belum berjalan optimal. Artinya, proses optimasi produksi perlu dilakukan untuk menghasilkan keunggulan kompetitif dalam menyediakan produk bermutu dan harga yang sesuai, apalagi *home* industri ‘Amanah’ memiliki siklus produksi aktif tiap hari.

Kondisi yang tidak jauh berbeda, penelitian yang pernah dilakukan Bagus Suryo A.U. (2010) misalnya, menggunakan "Aplikasi *Fuzzy Linear Programming* (FLP) untuk Mengoptimalkan Produksi Lampu: Studi Kasus di PT Sinar Terang Abadi". Penelitian tersebut bertujuan menggunakan metode *Fuzzy Linear Programming* untuk memaksimalkan produksi lampu di PT Surya Jaya Abadi atas kendala-kendala yang dihadapi seperti kapasitas mesin dan keterbatasan waktu. Untuk itu dibuat suatu model agar hasil produksi perusahaan optimal dengan tetap memperhatikan kendala produksi. Hasil penelitian menunjukkan bahwa penerapan *Fuzzy Linear Programming* mampu menghasilkan total produk sebesar 2033237 unit dengan nilai lamda sebesar λ= 0.897. Hasil ini lebih optimal, jika dibandingkan dengan *Linear Programming* biasa yang hanya menghasilkan produk sebesar 2029269 unit dan perusahaan dapat memenuhi semua tipe lampu yang disorder.

Mengingat berbagai kemungkinan buruk bisa saja terjadi selama proses produksi, maka sangat perlu bagi *home* industri ‘Amanah’ untuk melakukan optimasi produksi dengan metode *Fuzzy linear Programming*. FLP dipilih, karena telah banyak berhasil menangani masalah pengambilan keputusan dalam lingkungan yang semakin kompleks dan tidak pasti, akibat faktor subjektif atau intuitif (Bellman dan Zadeh, 1970).Teori ini juga dapat digunakan untuk menangani ketidakpastian dalam masalah dunia nyata, yang keanggotaannya dinyatakan dengan derajat keanggotaan tertentu dalam selang tertutup antara 0 dan 1 (Purba, 2012).

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana optimasi produksi *home* industri ‘Amanah’ dengan menggunakan metode *Fuzzy Linear Programming?*”

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dalam penelitian ini antara lain, menggunakan metode *Linear Programming* dan *Fuzzy Linear Programming* untuk menemukan optimasi produksi *home* industri ‘Amanah’. Serta mermbandingkan hasil optimasi antara *Linear Programming* dengan *Fuzzy Linear Programming*.

1.4. Batasan Masalah Penelitian

Batasan masalah dalam penelitian ini hanya menggunakan teori optimasi, program linier dan *fuzzy* dalam menyelesaikan masalah *Fuzzy Linear Programming* (FLP), untuk mencari fungsi tujuan (fungsi objektif) nilai optimal (maksimum) bahan baku produksi di *home* industri ‘Amanah’, kemudian *Fuzzy Linear Programming* (FLP) dibandingkan dengan *Linear Programming* biasa dalam mencari solusi produksi optimal.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah menambah ilmu pengetahuan di bidang *Linear Programming* serta bidang matematis, khususnya *Fuzzy Linear Programming*, agar dapat ditemukan solusi optimal produksi. Dapat memanfaatkan perhitungan *Fuzzy Linear Programming* untuk mencari solusi dan optimasi produksi. Menambah wawasan (*stock of knowledge*) mengenai *Linear Programming* khususnya *Fuzzy Linear Programming* dan untuk mengembangkannya lebih lanjut dalam bidang yang lebih luas.

BAB II
TINJAUAN PUSTAKA

2.1. Optimasi

Optimasi ialah suatu proses, cara, perbuatan mengoptimalkan, mencari yang paling menguntungkan, tertinggi dan terbaik (Kamus Besar Bahasa Indonesia, 2002).

Dalam disiplin matematika optimasi merujuk pada studi permasalahan yang mencoba mencari solusi optimal, yaitu penyelesaian yang tidak melanggar batasan-batasan yang ada yang paling mempunyai nilai tujuan terbesar atau terkecil, tergantung dari fungsi tujuannya yaitu maksimal atau minimal (Lieberman dan Hiller, 2001).

2.1.1. Nilai Optimal

Nilai optimal adalah nilai dari sebuah program linier dari sebuah fungsi tujuan yang bersesuaian dengan solusi optimalnya (Don, 2001). Model optimasi telah digunakan selama berabad-abad. Pada masa sekarang ini, optimasi menjadi sangat esensial untuk tujuan bisnis yang semakin kompleks dan rumit. Para insinyur pun menjadi semakin ambisius dalam mengembangkan hal ini. Dalam banyak hal, keputusan dapat saja dibuat tanpa mempertimbangkan tujuan dari model tersebut. Sebagai contoh, dalam kerjasama multinasional, sebagian kecil perkembangan proses operasi dapat mencapai peningkatan keuntungan berjuta-juta dolar. Tetapi, untuk mencapainya dibutuhkan analisis dan kerjasama setiap divisi.

Untuk model yang kompleks, dengan berbagai kerumitan yang ada, keputusan bisnis akan sangat berpengaruh. Dalam beberapa dasawarsa ini, telah dikembangkan *hardware* dan *software* komputer, yang berhasil melakukan optimasi secara praktis dalam bisnis dan ilmu pengetahuan. Sekarang ini, pemecahan masalah dengan ribuan atau bahkan jutaan variabel menjadi mungkin untuk diselesaikan.

2.1.2. Persoalan Optimasi dan Program Linier

Masalah optimasi adalah masalah memaksimumkan atau meminimumkan sebuah besaran tertentu yang disebut tujuan objektif (*objektive*) yang bergantung pada sejumlah berhingga variabel masukan (*input variables*). Variabel-variabel ini dapat tidak saling bergantung, atau saling bergantung melalui satu atau lebih kendala (*constrains*). Persoalan optimasi merupakan persoalan mencari nilai numerik terbesar (maksimasi) atau nilai numerik terkecil (minimasi) yang mungkin dari sebuah fungsi dari sejumlah variabel tertentu (Purba, 2012).

Dalam sebuah persoalan optimasi, adalah mencari nilai untuk variabel-variabel yang tidak melanggar (bertentangan) dengan kendala-kendala yang menyangkut variabel-variabel tersebut dan yang memberikan nilai optimum (maksimum atau minimum) pada fungsi yang hendak dioptimumkan itu. Biasanya kendala-kendala tersebut meliputi tenaga kerja, uang/modal, material yang merupakan *input* serta waktu dan ruang.

Persoalan *Program Linear* atau *Linear Program* ialah suatu persoalan untuk menentukan besarnya masing-masing nilai variabel sedemikian rupa sehingga nilai fungsi tujuan (*objektive function*) yang linear menjadi optimum (maksimum atau minimum) dengan memperhatikan pembatasan-pembatasan yang ada yaitu pembatasan mengenai masukannya ke dalam model matematik persamaan linear. Pembatasan-pembatasan ini pun harus dinyatakan dalam ketidaksamaan yang linier (*linear inequalities*).

Menurut Purba (2012), agar suatu masalah optimasi dapat diselesaikan dengan program linier, ada beberapa syarat atau karakteristik yang harus dipenuhi, yaitu:

1. Masalah tersebut harus dapat diubah menjadi permasalahan matematis. Ini berarti bahwa masalah tersebut harus bisa dituangkan ke dalam bentuk model matematik, dalam hal ini model linear, baik berupa persamaan maupun pertidaksamaan.
2. Adanya sasaran. Sasaran dalam model matematika masalah program linear berupa fungsi tujuan (fungsi objektif) yang akan dicari nilai optimalnya (maksimum/ minimum).
3. Ada tindakan alternatif, artinya nilai fungsi tujuan dapat diperoleh dengan berbagai cara dan diantaranya alternatif itu memberikan nilai optimal.
4. Sumber-sumber tersedia dalam jumlah yang terbatas (bahan mentah terbatas, modal terbatas, waktu terbatas, dll). Pembatasan-pembatasan harus dinyatakan di dalam ketidaksamaan yang linier (*linear inequalities*).
5. Keseluruhan sistem permasalahan harus dapat dipilah-pilah menjadi satuan-satuan aktivitas; sebagai misal: $a_{11}x_1 + a_{12}x_2 \leq k_1$, dimana x_1 dan x_2 adalah aktivitas.
6. Masing-masing aktivitas harus dapat ditentukan dengan tepat baik jenis maupun letaknya dalam model programasi.

7. Setiap aktivitas harus dapat dikuintifikasikan sehingga masing-masing nilainya dapat dihitung dan dibandingkan.
8. Koefisien model diketahui dengan pasti.
9. Bilangan yang digunakan dapat bernilai bulat/pecahan.
10. Semua variabel keputusan harus bernilai non negatif.

Program linier merupakan matematika terapan dari aljabar linier dimana dalam memecahkan persoalan dunia nyata melalui tahap-tahap sebagai berikut:

1. Menentukan aktivitas
2. Menentukan sumber-sumber (masukan)
3. Memahami masalah di bidang yang bersangkutan
4. Menghitung jumlah masukan dan keluaran untuk setiap satuan aktivitas.
5. Menentukan kendala-kendala aktivitas.
6. Menyusun atau merumuskan model matematika, yakni membentuk fungsi tujuan dan fungsi kendalanya.
7. Menyelesaikan model matematika (mencari jawaban model).
8. Menafsirkan jawaban model menjadi jawaban atas masalah yang nyata.

2.2. *Linear Programming*

Linear Programming merupakan model umum yang dapat digunakan dalam pemecahan masalah pengalokasian sumber-sumber yang terbatas secara optimal. Masalah tersebut timbul apabila seseorang diharuskan untuk memilih atau menentukan tingkat setiap kegiatan yang akan dilakukannya, di mana masing-masing kegiatan membutuhkan sumber yang sama sedangkan jumlahnya terbatas.

Pada masa modern sekarang, *Linear Programming* masih menjadi pilihan dalam upaya untuk memperoleh tingkat keuntungan maksimal atau biaya yang minimal. Dalam memecahkan masalah di atas, *Linear Programming* menggunakan model matematis. Sebutan "*linear*" berarti bahwa semua fungsi matematis yang disajikan dalam model ini haruslah fungsi-fungsi linier. Dalam *Linear Programming* dikenal dua macam fungsi, yaitu fungsi tujuan (*objective function*) dan fungsi-fungsi batasan (*constraint function*).

Fungsi tujuan adalah fungsi yang menggambarkan tujuan atau sasaran di dalam permasalahan *Linear Programming* yang berkaitan dengan pengaturan secara optimal sumber daya-sumber daya, untuk memperoleh keuntungan maksimal atau biaya minimal. Pada umumnya nilai yang akan dioptimalkan dinyatakan sebagai Z. Fungsi batasan merupakan bentuk penyajian secara matematis batasan-batasan kapasitas yang tersedia yang akan dialokasikan secara optimal ke berbagai kegiatan.

Menurut Supranto (1988), suatu persoalan disebut persoalan *Linear Programming* apabila memenuhi:

1. Tujuan (obyektif) yang akan dicapai harus dapat dinyatakan dalam fungsi linier. Fungsi ini disebut fungsi tujuan (fungsi obyektif).
2. Harus ada alternatif pemecahan yang membuat nilai fungsi tujuan optimum (laba yang maksimum, biaya yang minimum).
3. Sumber-sumber tersedia dalam jumlah yang terbatas (bahan mentah, modal, dan sebagainya). Kendala-kendala ini harus dinyatakan di dalam pertidaksamaan linier (*linear inequalities*).

Pada dasarnya, persoalan *Linear Programming* dapat dirumuskan sebagai berikut:

Cari $x_1, x_2, \dots, x_3, \dots, x_n$.

sedemikian rupa sehingga:

Maksimumkan $Z = c_1x_1 + c_2x_2 + \cdots + c_j x_j + \cdots + c_n x_n$ (2.1)

dengan batasan:

$$a_{11}x_1 + a_{21}x_2 + \cdots + a_{m1}x_m \leq b_1$$
$$a_{12}x_1 + a_{22}x_2 + \cdots + a_{m2}x_m \leq b_2$$
$$a_{13}x_1 + a_{23}x_2 + \cdots + a_{m3}x_m \leq b_3$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn} x_n \leq b_m$$
$$x_1, x_2, \dots, x_j, \dots, x_n \geq 0 \qquad (2.2)$$

Keterangan:

Ada n macam barang yang akan diproduksi masing-masing sebesar $x_1, x_2, \dots, x_j, \dots, x_n$

x_j = banyaknya produksi barang yang ke j, $j = 1,2, \dots, n$

c_j = harga per satuan barang ke j

Ada m macam bahan mentah masing-masing tersedia $b_1, b_2, \dots, b_j, \dots, b_n$

b_i = banyaknya bahan mentah ke i, $i = 1,2, \dots, m$

a_{ij} = banyaknya bahan mentah ke i yang dipergunakan untuk memproduksi 1 satuan barang ke j

x_j unit memerlukan a_{ij} unit bahan mentah i.

2.2.1. Asumsi-Asumsi *Linear Programming*

Asumsi-asumsi *Linear Programming* dapat dirinci sebagai berikut, yaitu:

1. *Proportionality*

   Asumsi ini berarti bahwa naik turunnya nilai Z dan penggunaan sumber atau fasilitas yang tersedia akan berubah secara sebanding (*proporsional*) dengan perubahan tingkat kegiatan.

   $$Z = c_1x_1 + c_2x_2 + c_3x_3 + \cdots c_nx_n \qquad (2.3)$$

   Setiap penambahan 1 unit x_1 akan menaikkan Z dengan c_1. Setiap penambahan 1 unit x_2 akan menaikkan Z dengan c_2, dan seterusnya.

   $$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_nx_n \leq b_1 \qquad (2.4)$$

   Setiap penambahan 1 unit x_1 akan menaikkan penggunaan sumber atau fasilitas 1 dengan a_{11}. Setiap penambahan 1 unit x_2 akan menaikkan penggunaan sumber atau fasilitas 1 dengan a_{12}, dan seterusnya. Asumsinya adalah, setiap ada kenaikan kapasitas riil tidak perlu ada biaya persiapan (*set up cost*).

2. *Additivity*

   Asumsi ini berarti bahwa nilai tujuan tiap kegiatan tidak saling mempengaruhi, atau dalam *linear programming* dianggap bahwa kenaikan dari nilai tujuan (Z) yang diakibatkan oleh kenaikan suatu kegiatan dapat ditambahkan tanpa mempengaruhi bagian nilai Z yang diperoleh dari kegiatan lain.

   $Z = 3x_1 + 5x_2$ di mana $x_1 = 10; x_2 = 2;$

   Sehingga $Z = 30 + 10 = 40$

   Jika x_1 bertambah 1 unit, maka sesuai dengan asumsi, maka nilai Z menjadi 40 + 3 = 43. Jadi, nilai 3 karena kenaikan x_1 dapat langsung ditambahkan pada nilai Z mula-

mula tanpa mengurangi bagian Z yang diperoleh dari kegiatan 2 (x_2). Dengan kata lain, tidak ada korelasi antara x_1 dan x_2.

3. *Divisibility*

   Asumsi ini menyatakan bahwa keluaran yang dihasilkan oleh setiap kegiatan dapat berupa bilangan pecahan. Demikian pula dengan nilai Z yang dihasilkan.

4. *Deterministic* (*certainty*)

   Asumsi ini menyatakan bahwa semua parameter yang terdapat dalam model *linear programming*(a_{ij}, b_i, c_j) dapat diperkirakan dengan pasti, meskipun jarang dengan tepat.

2.2.2. Prinsip Menggunakan *Simplex Linear Programming*

Apabila suatu masalah *Linear Programming* hanya mengandung dua kegiatan (variabel-variabel keputusan) saja, maka dapat diselesaikan dengan metode grafik. Bila terdapat lebih dari dua variabel maka metode grafik tidak dapat digunakan lagi, sehingga diperlukan metode simpleks. Metode ini lazim dipakai untuk menentukan kombinasi dari tiga variabel atau lebih.

Masalah *Linear Programming* yang melibatkan banyak variabel keputusan dapat dengan cepat dipecahkan dengan bantuan komputer. Bila variabel keputusan yang dikandung tidak terlalu banyak, masalah tersebut dapat diselesaikan dengan suatu algoritma yang biasanya sering disebut metode tabel simpleks. Disebut demikian karena kombinasi variabel keputusan yang optimal dicari dengan menggunakan tabel-tabel.

Menurut Subagyo dkk (1995), langkah-langkah metode tabel simpleks adalah sebagai berikut.

1. Mengubah fungsi tujuan dan batasan-batasan

   Fungsi tujuan diubah menjadi fungsi implisit, artinya semua c_jx_j digeser ke kiri

2. Menyusun persamaan-persamaan di dalam tabel

   Setelah formulasi disusun ke dalam tabel dan simbol, maka akan tampak seperti pada Tabel 2.1.

Tabel 2.1. Tabel Simpleks dalam Bentuk Simbol

Variabel dasar	Z	x_1	x_2	...	x_n	x_{n+1}	x_{n+2}	...	x_{n+m}	NK
Z	1	$-c_1$	$-c_2$	...	$-c_n$	0	0	...	0	0
x_{n+1}	0	a_{11}	a_{12}	...	a_{1n}	1	0	...	0	b_1
x_{n+2}	0	a_{21}	a_{22}	...	a_{2n}	0	1	...	0	b_2
.	.	.	.	...	.	.	.	...	.	.
.	.	.	.	...	.	.	.	...	.	.
x_{n+m}	0	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m

NK adalah nilai kanan persamaan, yaitu nilai di belakang tanda sama dengan (=). Variabel dasar adalah variabel yang nilainya sama dengan sisi kanan dari persamaan. Pada tabel tersebut, nilai variabel dasar pada fungsi tujuan (fungsi permulaan) ini harus 0, dan nilainya pada kendala-kendala bertanda positif.

Setelah data disusun dalam tabel-tabel di atas kemudian diadakan perubahan-perubahan agar dapat mencapai titik optimal, dengan langkah-langkah selanjutnya.

3. Memilih kolom kunci

   Kolom kunci adalah kolom yang merupakan dasar untuk mengubah tabel di atas. Dipilih kolom yang mempunyai nilai pada baris fungsi tujuan yang bernilai negatif dengan angka terbesar. Kalau suatu tabel sudah tidak memiliki nilai negatif pada baris fungsi tujuan, berarti tabel itu tidak bisa dioptimalkan lagi.

4. Memilih baris kunci

   Baris kunci adalah baris yang merupakan dasar untuk mengubah tabel tersebut di atas. Untuk itu, terlebih dahulu dicari indeks tiap-tiap baris dengan cara membagi nilai-nilai pada kolom NK dengan nilai yang sebaris pada kolom kunci.

   Indeks = nilai kolom NK / nilai kolom kunci. Kemudian dipilih angka yang memiliki nilai positif terkecil. (2.5)

5. Mengubah nilai-nilai baris kunci

   Nilai baris kunci diubah dengan cara membaginya dengan angka kunci.

6. Mengubah nilai-nilai selain pada baris kunci

   Nilai-nilai baris yang lain, selain pada baris kunci dapat diubah dengan rumus berikut:

   Baris baru = baris lama – (koefisien pada kolom kunci) x nilai baru baris kunci (2.6)
7. Melanjutkan perbaikan-perbaikan/perubahan-perubahan

   Ulangi langkah perbaikan mulai langkah ke-3 sampai langkah ke-6 untuk memperbaiki tabel-tabel yang telah diubah/diperbaiki nilainya. Perubahan baru berhenti setelah baris pertama (fungsi tujuan) tidak ada yang bernilai negatif.

Menurut Subagyo dkk (1995), terdapat beberapa ketentuan tambahan yang berupa :

1. Terdapat lebih dari satu kolom bernilai “negatif terbesar yang sama”

   Jika pada baris fungsi tujuan terdapat lebih dari satu kolom yang mempunyai nilai negatif yang angkanya terbesar dan sama, maka ada dua kolom yang bisa terpilih menjadi kolom kunci. Untuk mengatasi hal ini, dapat dipilih salah satu secara sembarang.
2. Dua baris atau lebih mempunyai indeks positif terkecil

   Jika ada dua baris atau lebih yang mempunyai nilai positif terkecil yang sama, maka ada beberapa baris yang dapat terpilih sebagai baris kunci. Dapat dipilih baris kunci secara bebas di antara keduanya dan hasilnya akan sama.
3. Kenaikan nilai Z tidak terbatas

   Nilai Z (tujuan) suatu permasalahan dapat ditambah terus bila paling tidak ada satu kegiatan yang tidak ada batasannya. Kalau di dalam *Linear Programming* muncul hal-hal semacam ini, tidak perlu dilanjutkan, cukup disebutkan bahwa kenaikan nilai Z dapat tidak terbatas. Di samping itu, ada baiknya pula bila diteliti lagi formulasi masalahnya, sebab hal ini dapat pula terjadi karena kesalahan dalam formulasi.
4. *Multiple Optional Solutions*

   Untuk mengetahui apakah suatu masalah *Linear Programming* bersifat *multiple solutions* atau tidak, dilihat baris fungsi tujuan pada tabel terakhir (optimal). Apabila dalam baris itu terdapat paling tidak satu kolom variabel yang mempunyai nilai 0 maka masalah itu bersifat *multiple solutions*. Masalah itu akan menghasilkan paling tidak dua alternatif yang mempunyai nilai Z yang sama.

Selain itu juga ada penyimpangan-penyimpangan dari bentuk standar, dimana penyimpangan-penyimpangan tersebut akan diatasi agar bisa diselesaikan dengan metode simpleks diantaranya adalah:

a. Batasan dengan tanda “sama dengan”

Jika suatu batasan memakai tanda kesamaan, maka cara mengatasinya dengan menambahkan variabel buatan (*artificial variable*).

b. Minimasi

Fungsi tujuan dari permasalahan *Linear Programming* yang bersifat minimasi, harus diubah menjadi maksimasi, agar sesuai dengan bentuk standar, yaitu maksimasi. Caranya adalah dengan mengganti tanda positif dan negatif pada fungsi tujuan.

c. Fungsi pembatas bertanda ≥

Bila suatu fungsi pembatas bertanda ≥, maka harus diubah menjadi ≤ dan akhirnya menjadi = agar dapat diselesaikan dengan metode simpleks.

d. Bagian kanan persamaan bertanda negatif

Bila bagian kanan persamaan bertanda negatif maka harus diubah menjadi positif. Caranya dengan mengubah tanda positif negatif dari tiap-tiap koefisien, kemudian ditambah dengan variabel buatan.

e. Bila minimum nilai x_j boleh negatif

Pada bentuk standar, nilai x_j harus selalu positif (dengan batasan $x_j \geq 0$). Tetapi kadang-kadang suatu masalah dapat menghasilkan formulasi *Linear Programming* yang memungkinkan nilai x_j negatif.

f. Bila nilai x_j boleh positif atau negatif

Jika hasil *Linear Programming* memungkinkan nilai x_j positif maupun negatif dan tidak ada batas negatif tertentu (negatif berapa pun dimungkinkan) maka nilai x_j diubah menjadi $x'_j - x''_j$, dengan ketentuan sebagai berikut:

x'_j = mewakili nilai positif dari x_j

x''_j = mewakili nilai negatif dari x_j

2.3. Logika *Fuzzy*

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*. Titik awal dari konsep modern mengenai ketidakpastian adalah *paper* yang dibuat oleh Lofti A Zadeh (1965), dimana Zadeh memperkenalkan teori yang memiliki obyek-obyek dari himpunan *fuzzy* yang memiliki batasan yang tidak presisi dan keanggotaan dalam himpunan *fuzzy*, dan bukan dalam bentuk logika benar (*true*) atau salah (*false*), tapi dinyatakan dalam derajat (*degree*). Konsep seperti ini disebut dengan *Fuzziness* dan teorinya dinamakan *Fuzzy Set Theory*.

Fuzziness dapat didefinisikan sebagai logika kabur berkenaan dengan semantik dari suatu kejadian, fenomena atau pernyataan itu sendiri. Seringkali ditemui dalam pernyataan yang dibuat oleh seseorang, evaluasi dan suatu pengambilan keputusan. Contoh:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayanan yang diberikan.
3. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.

Fuzzy system (sistem kabur) didasari atas konsep himpunan kabur yang memetakan domain *input* kedalam domain *output*. Perbedaan mendasar himpunan tegas dengan himpunan kabur adalah nilai keluarannya. Himpunan tegas hanya memiliki dua nilai output yaitu nol atau satu, sedangkan himpunan kabur memiliki banyak nilai keluaran yang dikenal dengan nilai derajat keanggotaannya.

2.3.1. Alasan Menggunakan Logika *Fuzzy*

Logika *fuzzy* adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian. Dimana logika klasik (*crisp*) menyatakan bahwa segala hal dapat diekspresikan dalam istilah binary (0 atau 1, hitam atau putih, ya atau tidak).

Logika *fuzzy* menggantikan kebenaran Boolean dengan tingkat kebenaran. Logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk linguistik, konsep tidak pasti seperti “sedikit”, “lumayan”, dan “sangat”. Logika ini diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Barkeley pada tahun 1965.

Logika *fuzzy* telah digunakan pada bidang-bidang seperti taksonomi, topologi, linguistik, teori automata, teori pengendalian, psikologi, *patternrecognition*, pengobatan, hukum, *decision analysis*, *system theory and information retrieval*. Pendekatan *fuzzy* memiliki kelebihan pada hasil yang terkait dengan sifat kognitif manusia, khususnya pada situasi yang melibatkan pembentukan konsep, pengenalan pola, dan pengambilan keputusan dalam lingkungan yang tidak pasti atau tidak jelas.

Ada beberapa alasan mengapa menggunakan logika *fuzzy* (Kusumadewi dkk., 2010) antara lain:

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang kompleks dan tidak pasti.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.3.2. Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA [x], memiliki 2 kemungkinan (Kusumadewi dkk, 2010) yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

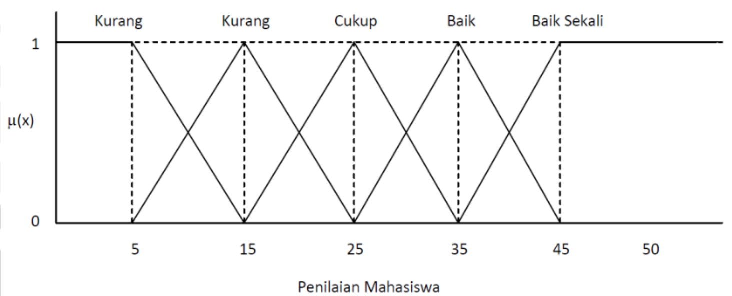
Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interprestasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Himpunan *fuzzy* memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: muda, parobaya, tua.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dan sebagainya.

2.3.3. Pemahaman *Fuzzy*

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Kusumadewi dkk, 2010), yaitu:

1. Variabel *Fuzzy*
   Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperatur, permintaan, dan sebagainya.
2. Himpunan *Fuzzy*
   Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.
   Contoh:
   a. Variabel mahasiswa, terbagi menjadi 5 himpunan *fuzzy*, yaitu: kurang sekali, kurang, cukup, baik dan baik sekali.
   b. Variabel dosen, terbagi menjadi 3 himpunan *fuzzy*, yaitu: cukup, baik, dan baik sekali. Seperti terlihat pada Gambar 2.1.



Gambar 2.1. Himpunan *Fuzzy* pada Variabel Mahasiswa

3. Semesta Pembicaraan

   Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

   Contoh:

   a. Semesta pembicaraan untuk variabel mahasiswa: (0 50)
   b. Semesta pembicaraan untuk variabel dosen: (0 50)

4. Domain

   Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif dan bilangan negatif.

   Contoh:

   1. Kurang sekali = (0 15)
   2. Kurang = (5 25)
   3. Cukup = (15 35)
   4. Baik = (25 45)
   5. Baik sekali = (35 50)

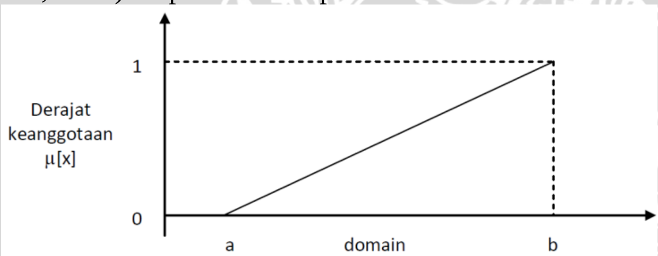
2.3.4. Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi *fuzzy* dalam U, maka A dapat dinyatakan sebagai pasangan terurut (Wulandari, F., 2005). Ada beberapa fungsi yang bisa digunakan.

2.3.4.1. Representasi *Linear*

Pada representasi *linear,* pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linier.

Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi (Kusumadewi dkk, 2010). Seperti terlihat pada Gambar 2.2.

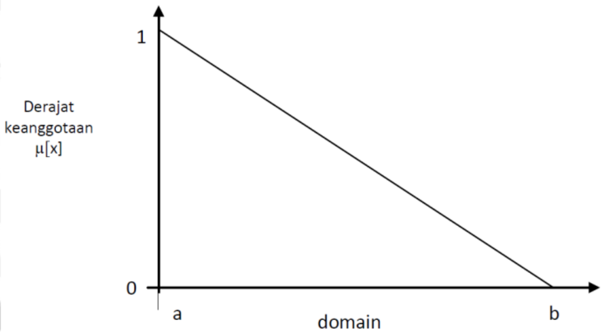


Gambar 2.2. Representasi Linier Naik

Fungsi keanggotaan:

$$\mu[x]\begin{cases} 0; & x \leq a \\ (x-a)/(b-a); & a < x < b \\ 1; & x \geq b \end{cases} \qquad (2.7)$$

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Seperti terlihat pada Gambar 2.3.

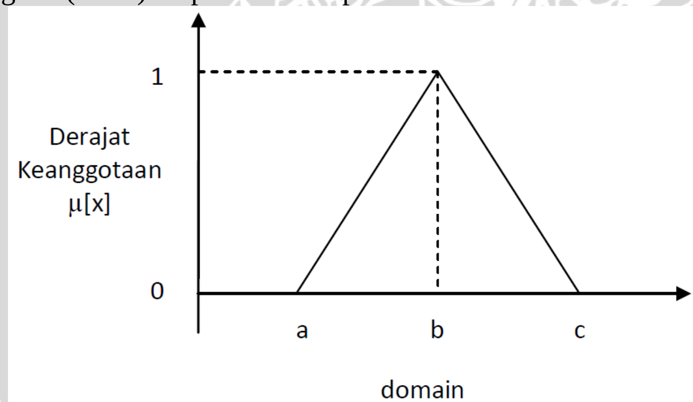


Gambar 2.3. Representasi Linier Turun
Fungsi keanggotaan:

$$\mu[x]\begin{cases} 1; & x \geq b \\ (b-x)/(b-a); & a < x < b \\ 0; & x \leq a \end{cases} \tag{2.8}$$

2.3.4.2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier). Seperti terlihat pada Gambar 2.4.

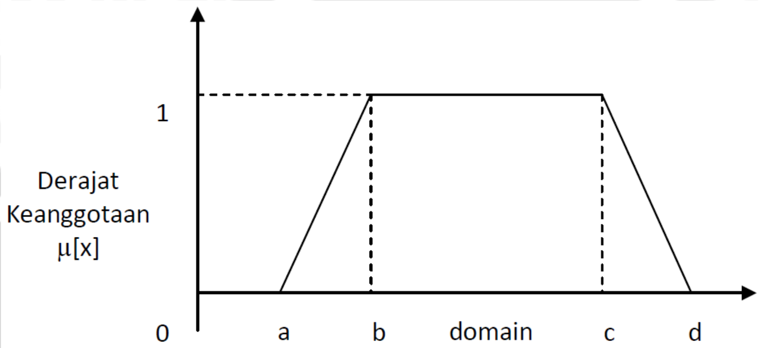


Gambar 2.4. Representasi Kurva Segitiga
Fungsi keanggotaan:

$$\mu[x]\begin{cases} 0; & x \geq c \; atau \; x \leq a \\ (x-a)/(b-a); & a < x < b \\ (c-x)/(c-b); & b < x < c \end{cases} \tag{2.9}$$

2.3.4.3. Representase Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Seperti terlihat pada Gambar 2.5.



Gambar 2.5. Representasi Kurva Trapesium

Fungsi keanggotaan:

$$\begin{cases} 0; & x \geq d \; atau \; x \leq a \\ (x-a)/(b-a); & a < x < b \\ (d-x)/(d-c); & c < x < d \\ 1; & b \leq x \leq c \end{cases} \qquad (2.10)$$

2.4. *Fuzzy Linear Programming*

Penyelesaian dengan *Fuzzy Linear Programming* (FLP), adalah pencarian suatu nilai Z yang merupakan fungsi obyektif yang akan dioptimasikan sedemikian rupa sehingga tunduk pada batasan-batasan yang dimodelkan dengan menggunakan himpunan *fuzzy* (Zimmermann, 1991). Dalam penjelasan selanjutnya hanya akan dibahas untuk persoalan maksimasi. Model matematika untuk persoalan maksimasi adalah sebagai berikut:

Tentukan x sedemikian hingga :

$$\begin{aligned} & \boldsymbol{c^T x \gtrapprox Z} \\ & \boldsymbol{Ax \lessapprox b} \\ & \boldsymbol{x \;\geq\; 0} \end{aligned} \qquad (2.11)$$

Dengan tanda '$\lessapprox$' merupakan bentuk *fuzzy* dari '≤' yang menginterpretasikan pada dasarnya kurang dari atau sama dengan.

Demikian pula, tanda '$\tilde{\geq}$' merupakan bentuk *fuzzy* dari '≥' yang menginterpretasikan pada dasarnya lebih dari atau sama dengan. Untuk kasus minimasi pada *Fuzzy Linear Programming*.

Tentukan x sedemikian hingga:

$$\boldsymbol{c^T x \tilde{\leq} Z}$$
$$\boldsymbol{Ax \tilde{\geq} b}$$
$$\boldsymbol{x \geq 0} \qquad (2.12)$$

Bentuk persamaan (2.11) dan (2.12) dapat dibawa ke dalam bentuk persamaan yaitu:

$$\boldsymbol{Bx \tilde{\leq} d}$$
$$\boldsymbol{x \geq 0} \qquad (2.13)$$

dengan :

$\boldsymbol{B} = \binom{-c}{A}$; dan $\boldsymbol{d} = \binom{-Z}{b}$; untuk kasus maksimasi, atau,

$\boldsymbol{B} = \binom{c}{-A}$; dan $\boldsymbol{d} = \binom{Z}{-b}$; untuk kasus minimasi. (2.14)

Tiap-tiap baris atau batasan (0, 1, 2, …, m) akan direpresentasikan dengan sebuah himpunan *fuzzy*, dengan fungsi keanggotaan pada himpunan ke-i adalah $\boldsymbol{\mu_i[B_i x]}$.

Fungsi keanggotaan untuk model keputusan himpunan *fuzzy* dapat dinyatakan sebagai berikut:

$$\boldsymbol{\mu_D[Bx]} = min\{\boldsymbol{\mu_i[B_i x]}\} \qquad (2.15)$$

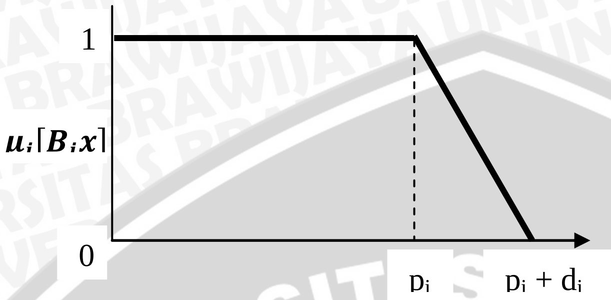
Tentu saja diharapkan akan mendapatkan solusi terbaik, yaitu suatu solusi dengan nilai keanggotaan yang paling besar, dengan demikian solusi yang sebenarnya adalah:

$$\max_{x\geq 0} \boldsymbol{\mu_D}\,[\boldsymbol{Bx}] = \max_{x\geq 0} \min_i\{\boldsymbol{\mu_i[B_i x]}\} \qquad (2.16)$$

dari sini terlihat bahwa $\boldsymbol{\mu_i[B_i x]} = 0$ jika batasan ke-i benar-benar dilanggar. Sebaliknya, $\boldsymbol{\mu_i[B_i x]} = 1$ jika batasan ke-i benar-benar dipatuhi. Nilai $\mu_i[\boldsymbol{B_i x}]$ akan naik secara monoton pada selang [0,1], yaitu:

$$\boldsymbol{\mu_i[B_i x]} = \begin{cases} 1; & jika\ \boldsymbol{B_i x} \leq d_i \\ \in [0,1]; & jika\ d_i < \boldsymbol{B_i x} < d_i + p_i \\ 0; & jika\ \boldsymbol{B_i x} \geq d_i + p_i \end{cases} \qquad (2.17)$$

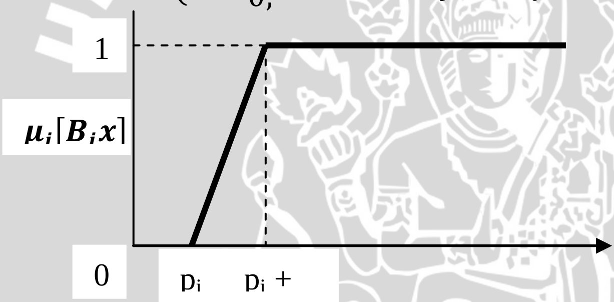
i= 0, 1,2,...,m



Gambar 2.6. Fungsi Keanggotaan Representasi Turun

Fungsi keanggotaan:

$$\boldsymbol{\mu_i}[\boldsymbol{B_i x}] = \begin{cases} 1; & jika\ \boldsymbol{B_i x} \leq d_i \\ 1 - \frac{\boldsymbol{B_i x} - d_i}{p_i}; & jika\ d_i < \boldsymbol{B_i x} < d_i + p_i \\ 0; & jika\ \boldsymbol{B_i x} \geq d_i + p_i \end{cases} \quad (2.18)$$



Gambar 2.7. Fungsi Keanggotaan Representasi Naik

Fungsi keanggotaan:

$$\boldsymbol{\mu_i}[\boldsymbol{B_i x}] = \begin{cases} 0; & jika\ \boldsymbol{B_i x} \leq d_i \\ 1 - \frac{(d_i + p_i) - \boldsymbol{B_i x}}{p_i}; & jika\ d_i < \boldsymbol{B_i x} < d_i + p_i \\ 1; & jika\ \boldsymbol{B_i x} \geq d_i + p_i \end{cases} \quad (2.19)$$

dengan p_i adalah toleransi interval yang diperbolehkan untuk melakukan pelanggaran baik pada fungsi obyektif maupun batasan. Dengan mensubstitusikan persamaan (2.18) ke (2.16) akan diperoleh:

$$\max_{x \geq 0} \boldsymbol{\mu_D}\ [\boldsymbol{Bx}] = \max_{x \geq 0} \min_i \left\{1 - \frac{\boldsymbol{B_i x} - d_i}{p_i}\right\} \quad (2.20)$$

Semakin besar nilai domain, akan memiliki nilai keanggotaan yang cenderung semakin kecil. Sehingga untuk mencari nilai *λ-cut* dapat dihitung sebagai λ=1-t, dengan:

$$d_i + p_i = \text{ruas kanan batasan ke-i} \quad (2.21)$$

Dengan demikian akan diperoleh bentuk *Linear Programming* baru sebagai berikut:

Maksimumkan: λ

Dengan batasan: $\lambda p_i + \boldsymbol{B_i x} \leq d_i + p_i$

$x \geq 0$

i = 0, 1, 2 ..., m (2.22)

(Kusumadewi dan Purnomo, 2010)

Menurut Li-Xin Wang (1997), terdapat tiga tipe dari *Fuzzy Linear Programming* :

- *Linear programming with fuzzy resources*

  Maksimumkan $\boldsymbol{cx}$

  Batasan $\boldsymbol{Ax} \mathrel{\widetilde{\leq}} \boldsymbol{b}$

  $\boldsymbol{x} \geq 0$ (2.23)

  Dengan yang difuzzikan adalah batasan ruas kanan. Dimana tanda $\widetilde{\leq}$ adalah merupakan bentuk *fuzzy* dari ≤.
- *Linear programming with fuzzy objective coefficients*

  Maksimumkan $\boldsymbol{\tilde{c}x}$

  Batasan $\boldsymbol{Ax} \leq \boldsymbol{b}$

  $\boldsymbol{x} \geq 0$ (2.24)

  Dengan yang difuzzikan adalah koefisien fungsi tujuan. Dimana $\tilde{c} = (\widetilde{c_1}, \dots, \widetilde{c_n})$adalah vektor dari bilangan *fuzzy*.
- *Linear programming with fuzzy constraint coefficients*

  Maksimumkan $\boldsymbol{cx}$

  Batasan $\boldsymbol{\tilde{A}x} \leq \boldsymbol{b}$

  $\boldsymbol{x} \geq 0$ (2.25)

  Dengan yang difuzzikan adalah koefisien batasan ruas kiri. Dimana $\tilde{A} = [\widetilde{a_{ij}}]$ adalah matriks yang terdiri dari bilangan *fuzzy*.

2.4.1. *Linear Programming with Fuzzy Resources*

Berdasarkan persamaan (2.23), biarkan $t_1(> 0)$ menjadi toleransi ke-i dari b_i, maka $(\boldsymbol{Ax})_i \lesssim b_i$ ditetapkan sebagai $(\boldsymbol{Ax})_i \lesssim b_i + \theta t$, dimana $\theta \in [0,1]$. Dengan kata lain $(\boldsymbol{Ax})_i \lesssim b_i$ didefinisikan sebagai himpunan *fuzzy* set i dengan fungsi keanggotaan :

$$\boldsymbol{\mu_i}(\boldsymbol{x}) = \begin{cases} 1 & jika \quad (\boldsymbol{Ax})_i < b_i \\ 1 - [(\boldsymbol{Ax})_i - b_i]/t_i & jika\ b_i \le (\boldsymbol{Ax})_i \le b_i + t_i \\ 0 & jika \quad (\boldsymbol{Ax})_i > b_i + t_i \end{cases} \quad (2.26)$$

Oleh karena itu, permasalahan untuk mencari x sedemikian rupa sehingga **cx** dan $\boldsymbol{\mu_i}(\boldsymbol{x})$ untuk i = 1, 2, ..., m dimaksimalkan. Ini adalah permasalahan optimasi *multiple objective*.

Werners (1987), memberikan metode berikut untuk memecahkan permasalah tersebut. Pertama memecahkan dua masalah *Linear Programming* berikut :

Maksimumkan $\boldsymbol{cx}$

Batasan $\quad (\boldsymbol{Ax})_i \le b_i,\ i = 1,2,\dots,m$

$$\boldsymbol{x} \ge 0 \quad (2.27)$$

Maksimumkan $\boldsymbol{cx}$

Batasan $(\boldsymbol{Ax})_i \le b_i + t_i,\ i = 1,2,\dots,m$

$$\boldsymbol{x} \ge 0 \quad (2.28)$$

Biarkan x^0 dan x^1 menjadi solusi persamaan (2.27) dan (2.28), dan menentukan $Z^0 = cx^0$ dan $Z^1 = cx^1$. Kemudian, fungsi keanggotaan berikut ini untuk mendefinisikan tingkat optimasi :

$$\boldsymbol{\mu_0}(\boldsymbol{x}) = \begin{cases} 1 & jika \quad \boldsymbol{cx} > Z^1 \\ 1 - \frac{Z^1 - cx}{Z^1 - Z^0} & jika\ Z^0 \le \boldsymbol{cx} \le Z^1 \\ 0 & jika \quad \boldsymbol{cx} < Z^0 \end{cases} \quad (2.29)$$

Sehingga, ketika $\boldsymbol{cx} \ge Z^1$ kita memiliki $\boldsymbol{\mu_0}(\boldsymbol{x}) = 1$, yang memberikan tingkat maksimum secara optimal, ketika $\boldsymbol{cx} \le Z^0$ kita memiliki $\boldsymbol{\mu_0}(\boldsymbol{x}) = 0$, yang memberikan tingkat minimum optimal, dan ketika **cx** terletak antara Z^1 dan Z^0 tingkat optimal berubah dari 1 ke 0.

Karena kendala dan fungsi tujuan diwakili oleh fungsi keanggotaan (2.26) dan (2.29), kita dapat menggunakan metode max-min untuk memecahkan masalah optimasi *multiple objective*. Sehingga permasalahannya menjadi :

$$\max_{x \ge 0} \min\ [\boldsymbol{\mu_0}(\boldsymbol{x}), \boldsymbol{\mu_1}(\boldsymbol{x}), \dots, \boldsymbol{\mu_m}(\boldsymbol{x})] \quad (2.30)$$

Atau ekuivalen

$$\begin{aligned} &\text{Maksimumkan} && \alpha \\ &\text{Batasan} && \boldsymbol{\mu_0}(\boldsymbol{x}) \geq \alpha \\ & && \boldsymbol{\mu_i}(\boldsymbol{x}) \geq \alpha, \ \ i = 1,2,\dots,m \\ & && \alpha \in [0,1], \ \ x \geq 0 \end{aligned} \tag{2.31}$$

Dengan mensubstitusikan persamaan (2.26) dan (2.29) ke persamaan (2.31), kita dapat menyimpulkan bahwa sumber permasalahan *Fuzzy Linear Programming* (2.23), dapat diselesaikan dengan memecahkan *Linear Programming* biasa berikut :

$$\begin{aligned} &\text{Maksimumkan } \alpha \\ &\text{Batasan} \quad \boldsymbol{cx} \geq Z^1 - (1-\alpha)(Z^1 - Z^0) \\ & \qquad\qquad (\boldsymbol{Ax})_{\boldsymbol{i}} \leq b_i + (1-\alpha)t_i, \ \ i = 1,2,\dots,m \\ & \qquad\qquad \alpha \in [0,1], \ \ x \geq 0 \end{aligned} \tag{2.32}$$

BAB III
METODE PENELITIAN

3.1. Data Penelitian

Pada penelitian ini data diperoleh dengan melakukan wawancara dengan pemilik secara langsung dan pengamatan di *home* industri ‘Amanah’. Data yang digunakan dalam penelitian ini adalah data produksi abon ayam *home* industri 'Amanah' untuk bahan baku dengan laba tiap produksi 300 ribu rupiah untruk krispi manis dan 450 ribu rupiah untuk krispi pedas.

Terdiri dari 2 tipe kategori data produksi untuk abon ayam krispi manis dan krispi pedas, yang belum dilakukan proses maksimasi/ optimasi (data mentah). Data tersebut dapat dilihat pada tabel 3.1 berikut:

Tabel 3.1. Data Bahan Baku Produksi

Bahan	Satuan unit		Jumlah bahan baku	Satuan
	Krispi manis	Krispi pedas		
Daging ayam	20	15	100	Kilogram
Gula	3	2,5	40	Kilogram
Bumbu	5	8	30	Kilogram
Cabe	1,5	2	8	Kilogram

3.2. Waktu dan Tempat

Pengamatan atau observasi ini dilakukan pada hari Senin tanggal 12 Agustus 2013. Pengamatan ini dilakukan di *home* industri ‘Amanah’ yang terletak di Jalan Wahid Hasyim no. 36 Desa Bringin Kecamatan Badas Kabupaten Kediri.

3.3. Metode Analisis

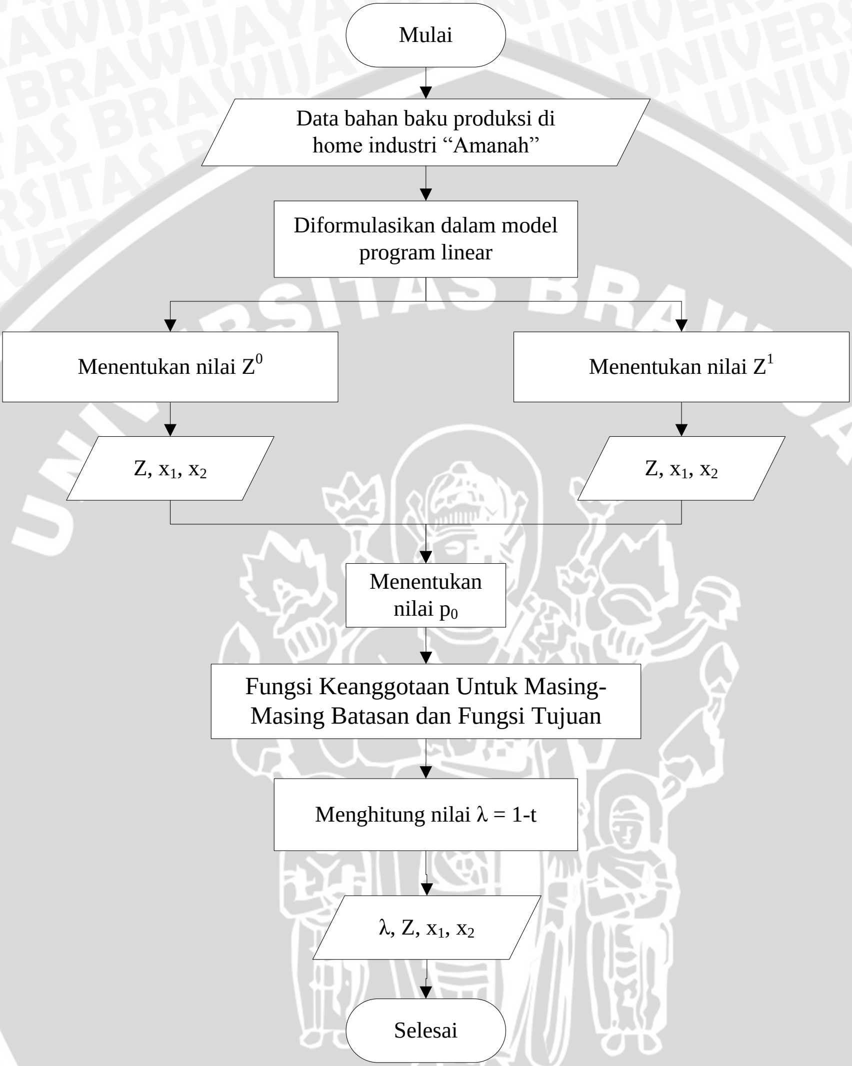
Dalam penelitian ini metode yang digunakan adalah *Fuzzy Linear Progamming* (FLP), untuk pencarian nilai Z yang merupakan fungsi obyektif yang akan dioptimasikan sedemikian rupa, sehingga tunduk pada batasan-batasan yang dimodelkan dengan menggunakan himpunan *fuzzy*.

Langkah-langkah analisis menggunakan metode *Fuzzy Linear Programming* adalah seperti berikut:

1. Menentukan model matematika sesuai dengan persamaan (2.1)
2. Mencari nilai Z^0 sesuai dengan persamaan (2.27) dan nilai Z^1 sesuai dengan persamaan (2.28)
3. Mencari nilai p_0
4. Menentukan fungsi keanggotaan untuk masing-masing batasan dan fungsi tujuan sesuai dengan persamaan (2.18) dan (2.19)
5. Menentukan nilai λ=1-t sesuai dengan persamaan (2.22)

Pada penyelesaian dengan persoalan *Fuzzy Linear Programming* sedikit berbeda dengan persoalan program linear biasa. Oleh sebab itu, penyelesaian dengan metode *Fuzzy Linear Programming* dilakukan kasus demi kasus dengan menggunakan metode simpleks dan dengan bantuan program WinQSB.

Tahapan dalam penerapan metode *Fuzzy Linear Programming* tersebut dapat digambarkan dalam diagram alir seperti pada Gambar 3.1.



Gambar 3.1. Diagram Alir Metode Penelitian

BAB IV
HASIL DAN PEMBAHASAN

4.1. Optimasi Produksi *Home* Industri ‘Amanah’ dengan Menggunakan Metode *Fuzzy Linear Programming*.

Home industri ‘Amanah’ adalah industri yang bergerak di bidang makanan jadi, yang memproduksi dua jenis abon ayam (krispi manis dan krispi pedas) dengan bahan-bahan mentah: daging ayam, gula, bumbu, dan cabe dengan laba tiap produksi 300 ribu rupiah untruk krispi manis dan 450 ribu rupiah untuk krispi pedas..

Namun demikian, pihak *home* industri ‘Amanah’ masih memungkinkan melakukan penambahan tiap bahan baku sampai dengan 10% dari tiap bahan baku yang ada. Dengan penambahan bahan baku sedikit saja, keuntungan yang diperoleh *home* industri akan bertambah. Kebutuhan bahan perjenis abon dan batas persediaan bahan baku untuk satu masa produksi serta besar laba dari penjualan untuk satu masa produksi tertera dalam Tabel 4.1 berikut:

Tabel 4.1. Bahan Baku Kebutuhan Produksi dan Toleransi

Bahan	Satuan unit		Kebutuhan bahan produksi		Satuan
	Krispi manis	Krispi pedas	Jumlah bahan baku	Toleransi (p_i)	
Daging ayam	20	15	100	10%*100 = 10 (p_1)	Kilogram
Gula	3	2,5	40	10%*40 = 4 (p_2)	Kilogram
Bumbu	5	8	30	10%*30 = 3 (p_3)	Kilogram
Cabe	1,5	2	8	10%*8 = 0,8 (p_4)	Kilogram

Variabel keputusan:
x_1: Jumlah krispi manis yang diproduksi
x_2: Jumlah krispi pedas yang diproduksi

4.1.1. Menentukan Model Matematika

Kasus tersebut dapat dimodelkan dalam model matematika sebagai berikut:

Maksimumkan : $Z = 10x_1 + 15x_2$

Batasan :

$$20x_1 + 15x_2 \leq 100$$
$$3x_1 + 2{,}5x_2 \leq 40$$
$$5x_1 + 8x_2 \leq 30$$
$$1{,}5x_1 + 2x_2 \leq 8$$
$$x_1, x_2 \geq 0$$

4.1.2. Mencari Nilai Z^0

Model matematika linear programming adalah:

Maksimumkan : $Z = 10x_1 + 15x_2$

Batasan :

$$20x_1 + 15x_2 \leq 100$$
$$3x_1 + 2{,}5x_2 \leq 40$$
$$5x_1 + 8x_2 \leq 30$$
$$1{,}5x_1 + 2x_2 \leq 8$$
$$x_1, x_2 \geq 0$$

Persoalan di atas, dapat diselesaikan dengan metode simpleks dalam bentuk standar program linier klasik (*Linear Programming*) berikut:

Maksimumkan: $Z = 10x_1 + 15x_2$

Batasan : $20x_1 + 15x_2 + S_1 = 100$

$$3x_1 + 2{,}5x_2 + S_2 = 40$$
$$5x_1 + 8x_2 + S_3 = 30$$
$$1{,}5x_1 + 2x_2 + S_4 = 8$$
$$x_1, x_2 \geq 0$$

Tabel 4.2. Tabel Simpleks untuk Solusi Awal

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi	
Z	1	-10	-15	0	0	0	0	0	
S_1	0	20	15	1	0	0	0	100	6,666667
S_2	0	3	2,5	0	1	0	0	40	16
S_3	0	5	8	0	0	1	0	30	3,75
S_4	0	1,5	2	0	0	0	1	8	4

Variabel masuk : x_2
Variabel keluar : S_3

Tabel 4.3. Tabel Simpleks untuk Solusi Baru

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi	
Z	1	-0,625	0	0	0	1,875	0	56,25	
S_1	0	10,625	0	1	0	-1,875	0	43,75	4,117647
S_2	0	1,4375	0	0	1	-0,3125	0	30,625	21,30435
x_2	0	0,625	1	0	0	0,125	0	3,75	6
S_4	0	0,25	0	0	0	-0,25	1	0,5	2

Variabel masuk : x_1
Variabel keluar : S_4

Tabel 4.4. Tabel Simpleks untuk Solusi Akhir

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi
Z	1	0	0	0	0	1,25	2,5	57,5
S_1	0	0	0	1	0	8,75	-42,5	22,5
S_2	0	0	0	0	1	1,125	-5,75	27,75
x_2	0	0	1	0	0	0,75	-2,5	2,5
x_1	0	1	0	0	0	-1	4	2

Karena semua nilai pada baris Z pada tabel solusi akhir sudah positif atau nol, maka tabel solusi akhir merupakan tabel optimal. Dari Tabel 4.4 dan juga perhitungan dengan bantuan program WinQSB (lampiran 1) dapat diperoleh hasil akhir adalah sebagai berikut:

$$Z = 57{,}5$$
$$x_1 = 2$$
$$x_2 = 2{,}5$$

4.1.3. Mencari Nilai Z[1]

Model awal *Linear Programming* dapat diubah menjadi:

Maksimumkan : $Z = 10x_1 + 15x_2$

Dengan batasan :

$$20x_1 + 15x_2 \leq 110$$
$$3x_1 + 2{,}5x_2 \leq 44$$
$$5x_1 + 8x_2 \leq 33$$
$$1{,}5x_1 + 2x_2 \leq 8{,}8$$
$$x_1, x_2 \geq 0$$

Persoalan tersebut, dapat diselesaikan dengan menggunakan metode simpleks bentuk standar *Linear Programming*:

Maksimumkan : $Z = 10x_1 + 15x_2$

Dengan batasan :

$$20x_1 + 15x_2 + S_1 = 110$$
$$3x_1 + 2{,}5x_2 + S_2 = 44$$
$$5x_1 + 8x_2 + S_3 = 33$$
$$1{,}5x_1 + 2x_2 + S_4 = 8{,}8$$
$$x_1, x_2 \geq 0$$

Tabel 4.5. Tabel Simpleks untuk Solusi Awal

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi	
Z	1	-10	-15	0	0	0	0	0	
S_1	0	20	15	1	0	0	0	110	7.333333
S_2	0	3	2,5	0	1	0	0	44	17,6
S_3	0	5	8	0	0	1	0	33	4,125
S_4	0	1,5	2	0	0	0	1	8,8	4,4

Variabel masuk : x_2

Variabel keluar : S_3

Tabel 4.6. Tabel Simpleks untuk Solusi Baru

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi	
Z	1	-0,625	0	0	0	1,875	0	61,875	
S_1	0	10,625	0	1	0	-1,875	0	48,125	4,529412
S_2	0	1,4375	0	0	1	-0,3125	0	33,6875	23,43478
x_2	0	0,625	1	0	0	0,125	0	4,125	6,6
S_4	0	0,25	0	0	0	-0,25	1	0,55	2,2

Variabel masuk : x_1
Variabel keluar : S_4

Tabel 4.7. Tabel Simpleks untuk Solusi Akhir

	Z	x_1	x_2	S_1	S_2	S_3	S_4	Solusi
Z	1	0	0	0	0	1,25	2,5	63,25
S_1	0	0	0	1	0	8,75	-42,5	24,75
S_2	0	0	0	0	1	1,125	-5,75	30,525
x_2	0	0	1	0	0	0,75	-2,5	2,75
x_1	0	1	0	0	0	-1	4	2,2

Karena semua nilai pada baris Z pada tabel solusi akhir sudah positif atau nol, maka tabel solusi akhir merupakan tabel optimal. Dari Tabel 4.7 dan juga perhitungan dengan bantuan program WinQSB (lampiran 2) dapat diperoleh hasil akhir adalah sebagai berikut:

$$Z = 63,25$$
$$x_1 = 2,2$$
$$x_2 = 2,75$$

4.1.4. Mencari Nilai p_0

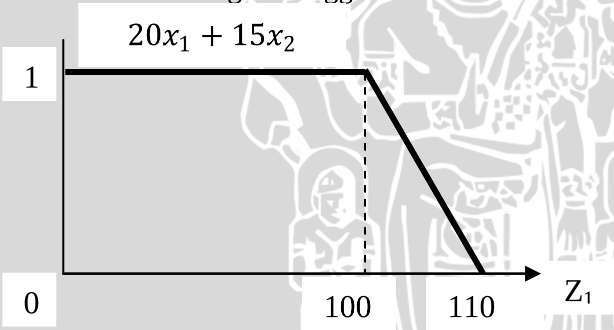
Dari kedua hasil ini Z^0 dan Z^1, kita dapat menentukan nilai p_0, yaitu hasil pengurangan dari Z^1 dengan Z^0 ($p_0 = 63{,}25 - 57{,}5 = 5{,}75$).

Tabel 4.8. Batasan *Non*-Fuzzy dan Batasan *Fuzzy*

	Batasan *Non-Fuzzy*	Batasan *Fuzzy*	
		t=0	t=1
Fungsi objektif		57,5	63,25
Batasan 1	100	100	110
Batasan 2	40	40	44
Batasan 3	30	30	33
Batasan 4	8	8	8,8

4.1.5. Fungsi Keanggotaan untuk Masing-Masing Batasan dan Fungsi Tujuan

Berikut adalah fungsi keanggotaan untuk batasan 1.

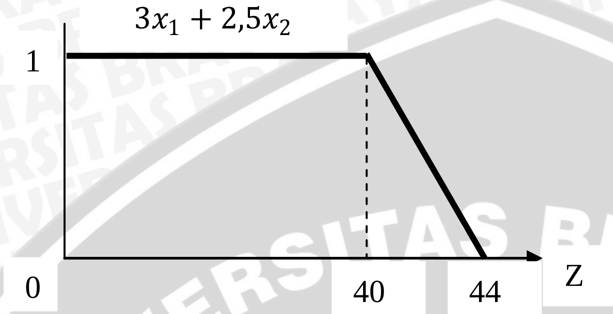


Gambar 4.1. Fungsi Keanggotaan Batasan 1

dengan,

$$\mu(Z_1) = \begin{cases} 1; & jika\ Z_1 \le 100 \\ 1 - \dfrac{Z_1 - 100}{10}; & jika\ 100 < Z_1 < 110 \\ 0; & jika\ Z_1 \ge 110 \end{cases}$$

Berikut adalah fungsi keanggotaan untuk batasan 2.

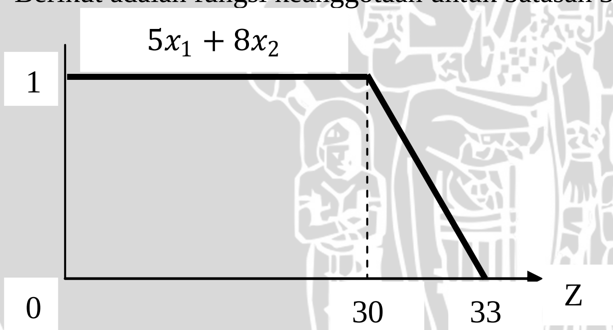


Gambar 4.2. Fungsi Keanggotaan Batasan 2

dengan,

$$\mu(Z_2) = \begin{cases} 1; & jika\ Z_2 \leq 40 \\ 1 - \dfrac{Z_2 - 40}{4}; & jika\ 40 < Z_2 < 44 \\ 0; & jika\ Z_2 \geq 44 \end{cases}$$

Berikut adalah fungsi keanggotaan untuk batasan 3.

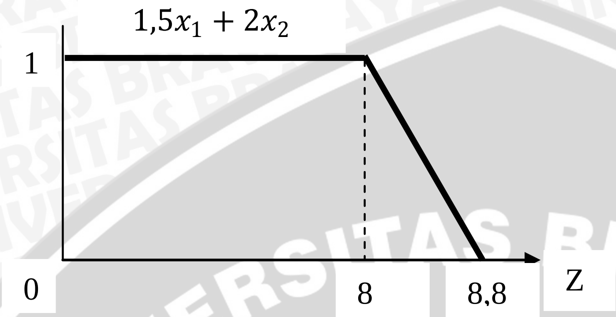


Gambar 4.3. Fungsi Keanggotaan Batasan 3

dengan,

$$\mu(Z_3) = \begin{cases} 1; & jika\ Z_3 \leq 30 \\ 1 - \dfrac{Z_3 - 30}{3}; & jika\ 30 < Z_3 < 33 \\ 0; & jika\ Z_3 \geq 33 \end{cases}$$

Berikut adalah fungsi keanggotaan untuk batasan 4.

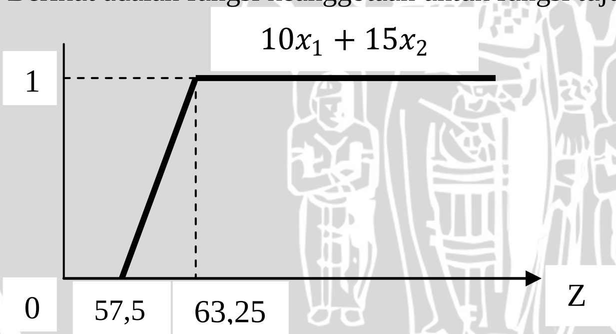


Gambar 4.4. Fungsi Keanggotaan Batasan 4

dengan,

$$\mu(Z_4) = \begin{cases} 1; & jika\ Z_4 \leq 8 \\ 1 - \dfrac{Z_4 - 8}{0{,}8}; & jika\ 8 < Z_4 < 8{,}8 \\ 0; & jika\ Z_4 \geq 8{,}8 \end{cases}$$

Berikut adalah fungsi keanggotaan untuk fungsi tujuan.



Gambar 4.5. Fungsi Keanggotaan untuk Fungsi Tujuan

dengan,

$$\mu(Z_5) = \begin{cases} 0; & jika\ Z_5 \leq 57{,}5 \\ 1 - \dfrac{63{,}25 - Z_5}{5{,}75}; & jika\ 57{,}5 < Z_5 < 63{,}25 \\ 1; & jika\ Z_5 \geq 63{,}5 \end{cases}$$

4.1.6. Menghitung Nilai λ=1-t

Akhirnya dapat dibentuk model *Fuzzy Linear Programming* sebagai berikut:

Maksimumkan : λ

Dengan batasan :

$$5{,}75\lambda - (10x_1 + 15x_2) \leq -63{,}25 + 5{,}75 = -57{,}5$$
$$10\lambda + 20x_1 + 15x_2 \leq 100 + 10 = 110$$
$$4\lambda + 3x_1 + 2{,}5x_2 \leq 40 + 4 = 44$$
$$3\lambda + 5x_1 + 8x_2 \leq 30 + 3 = 33$$
$$0{,}8\lambda + 1{,}5x_1 + 2x_2 \leq 8 + 0{,}8 = 8{,}8$$
$$\lambda, x_1, x_2 \geq 0$$

Sehingga bentuk *Linear Programming* menjadi:

Maksimum : λ

Dengan batasan :

$$-5{,}75\lambda + 10x_1 + 15x_2 \geq 57{,}5$$
$$10\lambda + 20x_1 + 15x_2 \leq 110$$
$$4\lambda + 3x_1 + 2{,}5x_2 \leq 44$$
$$3\lambda + 5x_1 + 8x_2 \leq 33$$
$$0{,}8\lambda + 1{,}5x_1 + 2x_2 \leq 8{,}8$$
$$\lambda, x_1, x_2 \geq 0$$

Bentuk standar *Linear Programming* menjadi:

Maksimum : $Z = \lambda$

Dengan batasan :

$$-5{,}75\lambda + 10x_1 + 15x_2 - S_1 + R_1 = 57{,}5$$
$$10\lambda + 20x_1 + 15x_2 + S_2 = 110$$
$$4\lambda + 3x_1 + 2{,}5x_2 + S_3 = 44$$
$$3\lambda + 5x_1 + 8x_2 + S_4 = 33$$
$$0{,}8\lambda + 1{,}5x_1 + 2x_2 + S_5 = 8{,}8$$
$$\lambda, x_1, x_2, S_1, S_2, S_3, S_4, S_5 \geq 0$$

Linear Programming ini harus diselesaikan dengan dua tahap, yaitu:

Tahap 1

Menyelesaikan *Linear Programming*:

Min : r = R_1

Dengan batasan :

$$-5{,}75\lambda + 2x_1 + 3x_2 - S_1 + R_1 = 57{,}5$$
$$10\lambda + 20x_1 + 15x_2 + S_2 = 110$$
$$4\lambda + 3x_1 + 2{,}5x_2 + S_3 = 44$$
$$3\lambda + 5x_1 + 8x_2 + S_4 = 33$$
$$0{,}8\lambda + 1{,}5x_1 + 2x_2 + S_5 = 8{,}8$$
$$\lambda, x_1, x_2, S_1, S_2, S_3, S_4, S_5 \geq 0$$

Diperoleh variabel *basic*: R_1, S_2, S_3, S_4. Karena R_1 muncul di persamaan *r*, maka harus disubstitusikan dengan batasan pertama.

$$R_1 = 57{,}5 + 5{,}75\lambda - 10x_1 - 15x_2 + S_1$$

Dengan mensubstitusikan R_1 ke persamaan r, maka *Linear Programming* yang harus diselesaikan adalah:

Min: $r = 57{,}5 + 5{,}75\lambda - 10x_1 - 15x_2 + S_1$

Dengan batasan:

$$-5{,}75\lambda + 10x_1 + 15x_2 - S_1 + R_1 = 57{,}5$$
$$10\lambda + 20x_1 + 15x_2 + S_2 = 110$$
$$4\lambda + 3x_1 + 2{,}5x_2 + S_3 = 44$$
$$3\lambda + 5x_1 + 8x_2 + S_4 = 33$$
$$0{,}8\lambda + 1{,}5x_1 + 2x_2 + S_5 = 8{,}8$$
$$\lambda, x_1, x_2, S_1, S_2, S_3, S_4, S_5 \geq 0$$

Tabel 4.9. Tabel Simpleks untuk Solusi Awal

	r	λ	x_1	x_2	S_1	S_2	S_3	S_4	S_5	R_1	Solusi	
r	1	-5,75	10	15	-1	0	0	0	0	0	57,5	
R_1	0	-5,75	10	15	-1	0	0	0	0	1	57,5	3,833333
S_2	0	10	20	15	0	1	0	0	0	0	110	7,333333
S_3	0	4	3	2,5	0	0	1	0	0	0	44	17,6
S_4	0	3	5	8	0	0	0	1	0	0	33	4,125
S_5	0	0,8	1,5	2	0	0	0	0	1	0	8,8	4,4

Variabel masuk : x_2

Variabel keluar : R_1

Tabel 4.10. Tabel Simpleks untuk Solusi Baru

	r	λ	x_1	x_2	S_1	S_2	S_3	S_4	S5	R_1	Solusi
r	1	0	0	0	0	0	0	0	0	-1	0
x_2	0	-0,38333	0,666667	1	-0,06667	0	0	0	0	0,066667	3,833333
S_2	0	15,75	10	0	1	1	0	0	0	-1	52,5
S_3	0	4,958333	1,333333	0	0,166667	0	1	0	0	-0,16667	34,41667
S_4	0	6,066667	-0,33333	0	0,533333	0	0	1	0	-0,53333	2,333333
S_5	0	1,566667	0,166667	0	0,133333	0	0	0	1	-0,13333	1,133333

Tahap 2

Menyelesaikan *Linear Programming*:

Maks: $z = \lambda$

Tabel 4.11. Tabel Simpleks untuk Solusi Awal

	Z	λ	x_1	x_2	S_1	S_2	S_3	S_4	S_5	Solusi	
Z	1	-1	0	0	0	0	0	0	0	0	
x_2	0	-0,38333	0,666667	1	-0,06667	0	0	0	0	3,833333	
S_2	0	15,75	10	0	1	1	0	0	0	52,5	3,333333
S_3	0	4,958333	1,333333	0	0,166667	0	1	0	0	34,41667	6,941176
S_4	0	6,066667	-0,33333	0	0,533333	0	0	1	0	2,333333	0,384615
S_5	0	1,566667	0,166667	0	0,133333	0	0	0	1	1,133333	0,723404

Variabel masuk : λ

Variabel keluar : S_4

Tabel 4.12. Tabel Simpleks untuk Solusi Baru

	Z	λ	x_1	x_2	S_1	S_2	S_3	S_4	S_5	Solusi	
Z	1	0	-0,05495	0	0,087912	0	0	0,164835	0	0,384615	
x_2	0	0	0,645604	1	-0,03297	0	0	0,063187	0	3,980769	
S_2	0	0	10,86538	0	-0,38462	1	0	-2,59615	0	46,44231	4,274336
S_3	0	0	1,605769	0	-0,26923	0	1	-0,81731	0	32,50962	20,24551
λ	0	1	-0,05495	0	0,087912	0	0	0,164835	0	0,384615	
S_5	0	0	0,252747	0	-0,0044	0	0	-0,25824	1	0,530769	2,1

Variabel masuk : x_1

Variabel keluar : S_5

Tabel 4.13. Tabel Simpleks untuk Solusi Akhir

	Z	λ	x_1	x_2	S_1	S_2	S_3	S_4	S_5	Solusi
Z	1	0	0	0	0,002899	0	0	0,108696	0,217391	0,5
x_2	0	0	0	1	-0,00072	0	0	0,722826	-2,55435	2,625
S_2	0	0	0	0	-0,00652	1	0	8,505435	-42,9891	23,625
S_3	0	0	0	0	-0,00804	0	1	0,82337	-6,35326	29,1375
λ	0	1	0	0	0,002899	0	0	0,108696	0,217391	0,5
x_1	0	0	1	0	-0,00058	0	0	-1,02174	3,956522	2,1

Maka hasil akhir yang diperoleh dari Tabel 4.13 dan juga perhitungan dengan bantuan program WinQSB (lampiran 3) adalah:

$$\lambda = 0,5$$
$$x_1 = 2,1$$
$$x_2 = 2,625$$
$$Z = 10x_1 + 15x_2 = 60,375$$

Berdasarkan hasil akhir di atas, maka solusi *Linear Programming* pada kasus *Non-Fuzzy* dan *Fuzzy* terlihat pada tabel 4.14 berikut.

Tabel 4.14. Solusi *Non-Fuzzy* dan *Fuzzy*

Solusi *non-fuzzy*	Solusi *fuzzy*
$x_1 = 2$ $x_2 = 2,5$ $Z = 57,5$	$x_1 = 2,1$ $x_2 = 2,625$ $Z = 60,375$

4.2. Interpretasi

Dengan menggunakan *Linear Programming* biasa, keuntungan maksimum akan diperoleh jika produk krispi manis diproduksi sebanyak 2 kali produksi dan produk krispi pedas sebanyak 2,5 kali produksi, keuntungan yang diperoleh sebesar Rp. 57.500,- dalam sehari. Pada kondisi ini, dibutuhkan daging ayam sebanyak $77{,}5\ (20 * 2 + 15 * 2{,}5)$ kg; gula sebanyak $12{,}25\ (3 * 2 + 2{,}5 * 2{,}5)$ kg; bumbu sebanyak $30\ (5 * 2 + 8 * 2{,}5)$ kg; cabe sebanyak $8\ (1{,}5 * 2 + 2 * 2{,}5)$ kg.

Apabila digunakan *Fuzzy Linear Programming* keuntungan maksimum akan diperoleh jika produk krispi manis diproduksi sebanyak 2,1 kali produksi dan produk krispi pedas sebanyak 2,625 kali produksi, keuntungan yang diperoleh sebesar Rp. 60.375,- (Rp. 2.875,- lebih banyak dibanding dengan *Linear Programming* biasa) dalam sehari. Dengan catatan bahwa pada kondisi ini dibutuhkan daging ayam sebanyak $81{,}375\ (20 * 2{,}1 + 15 * 2{,}625)$ kg; gula sebanyak $12{,}86\ (3 * 2{,}1+2{,}5*2{,}625$ kg; bumbu sebanyak $31{,}5\ 5*2{,}1+8*2{,}625$ kg; cabe sebanyak $8{,}4\ (1{,}5 * 2{,}1 + 2 * 2{,}625)$ kg.

Nilai $\lambda = 0{,}5$ mengandung pengertian bahwa $\lambda = 1 - t$ untuk setiap himpunan yang digunakan untuk mengimplementasi setiap batasan sebesar 0,5. Dengan kata lain, skala terbesar $t = 1 - 0{,}5 = 0{,}5$ digunakan untuk menentukan besarnya penambahan terbesar dari setiap batasan dari setiap batasan yang diizinkan. Pada batasan ke-1, penambahan daging ayam diizinkan hingga 110 kg, pada kenyataannya penambahan yang dibutuhkan maksimal hanya sebesar $0{,}5 * 110\ \text{kg} = 55\ \text{kg}$. Pada batasan ke-2, penambahan gula diizinkan hingga 40 kg, pada kenyataanya penambahan yang dibutuhkan maksimal hanya sebesar $0{,}5 * 44\ \text{kg} = 22\ \text{kg}$. Pada batasan ke-3, penambahan bumbu diizinkan hingga 33 kg, pada kenyataanya penambahan yang dibutuhkan maksimal hanya sebesar $0{,}5 * 33\ \text{kg} = 16{,}5\ \text{kg}$. Pada batasan ke-4, penambahan bumbu diizinkan hingga 8,8 kg, pada kenyataanya penambahan yang dibutuhkan maksimal hanya sebesar $0{,}5 * 8{,}8\ \text{kg} = 4{,}4\ \text{kg}$.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dengan menggunakan hasil penyelesaian di atas, maka dapat ditarik kesimpulan:

1. Menggunakan *Linear Programming* biasa, keuntungan maksimum yang dapat diterima oleh *home* industri ‘Amanah’ dalam memproduksi abon ayam sebesar Rp. 57.500,-/hari dengan harus memproduksi abon ayam krispi manis sebanyak 2 kali produksi dan abon ayam krispi pedas sebanyak 2 kali produksi dalam sehari.
2. Menggunakan *Fuzzy Linear Programming*, keuntungan maksimum yang dapat diterima oleh *home* industri ‘Amanah’ dalam memproduksi abon ayam adalah sebesar Rp. 60.375,-/hari (Rp. 2.875,- lebih banyak dibanding dengan *Linear Programming* biasa) dengan harus memproduksi abon ayam krispi manis sebanyak 2 kali produksi dan abon ayam krispi pedas sebanyak 3 kali produksi dalam sehari.
3. Penyelesaian *Fuzzy Linear Programming* akan memberikan hasil lebih baik dan optimal jika dibandingkan dengan penyelesaian *Linear Programming* biasa.

5.2. Saran

Berdasarkan kesimpulan di atas, maka saran yang dapat diberikan adalah sebaiknya menggunakan *Fuzzy Linear Programming* dalam pengambilan keputusan dari permasalahan program linier yang kompleks, terutama dalam upaya optimasi produksi di sebuah perusahaan atau instansi. Sebab program linier biasa tidak melibatkan asumsi-asumsi yang selalu terbentuk dalam dunia nyata.

Hasil penelitian ini dapat digunakan sebagai referensi utama untuk melakukan penelitian selanjutnya, tentang optimasi produksi dengan menggunakan metode *Fuzzy Linear Programming*.

DAFTAR PUSTAKA

Don, T. 2001. *Fundamental of Network Analysis*. New Jersey: Prentice Hall International Series In Industrial and System Engineering.

Bellman, R.E. dan L.A. Zadeh. 1970. *Decision Making in a Fuzzy Environment*. Management Sciences 17(4): 141-164.

Fisiana, F. 2010. *Optimasi Biaya Produksi Menggunakan Metode Revised Multi Choice Goal Programming dengan Tahap Persediaan Terkontrol Supply Chain Model: Studi Kasus di PT. Gunungarta Manunggal, Pasuruan*. Jurnal Matematika, 12 (2): 1-13.

Kusumadewi, S. dan H. Purnomo. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.

Lieberman and Hiller. 2001. *Introduction to Operation Research. Fifth Edition*. Diterjemahkan oleh Ellen Gunawan & Ardi Wirda Mulia. Jakarta: Penerbit Erlangga.

Purba, R. 2012. *Penerapan Logika Fuzzy Pada Program Linear*. Jurnal Matematika, 10 (11): 3-6

Subagyo, P.; M. Asri; dan H. Handoko. 1995. *Dasar-Dasar Operations Research*. Yogyakarta: BPFE Yogyakarta.

Supranto, J. 1988. *Riset Operasi: Untuk Pengambilan Keputusan*. Jakarta: Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia.

Susanto, T. dan Sarwadi, 2006. *Optimasi Produksi dan Pengendalian Bahan Baku Studi Kasus Pada PT. Joshua Indo Export*. Jurnal Matematika, 9 (1): 133-138

Utomo, B. S. A. 2010. *Aplikasi Fuzzy Linear Programming Untuk Mengoptimalkan Produksi Lampu: Studi Kasus di PT. Sinar Terang Abadi*. Jurnal Sistem Teknik Industri, 7 (1): i.

Wang, L. X. 1997. *A Course in Fuzzy Systems and Control*. Singapore: Prentice Hall International.

Werners, B. 1987. *An Interactive Fuzzy Programming System*. Fuzzy Sets and Systems, 23: 131-147.

Wulandari, F. 2005. *Pembuatan Sistem Pendukung Keputusan Berbasis Teori Fuzzy Untuk Mengembangkan Suatu Produk Baru*. Jurnal Sain, Teknologi & Industri, 2 (5): 62-66.

Zimmermann. 1991. *Fuzzy Set Theory an Its Applications*. Massachusetts: Kluwer Academic Publishers.

Lampiran 1. Langkah Perhitungan untuk Mencari Z^0 Menggunakan Program WinQSB

Memasukkan data

	OBJ/Constraint/Bound
Maximize	10x1+15x2
C1	20x1+15x2<=100
C2	3x1+2.5x2<=40
C3	5x1+8x2<=30
C4	1.5x1+2x2<=8
Integer:	
Binary:	
Unrestricted:	
X1	>=0, <=M
X2	>=0, <=M

Solve and Analyze → Solve and Display Steps.

Iterasi 1

Simplex Tableau -- Iteration 1

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	20,0000	15,0000	1,0000	0	0	0	100,0000	6,6667
Slack_C2	0	3,0000	2,5000	0	1,0000	0	0	40,0000	16,0000
Slack_C3	0	5,0000	8,0000	0	0	1,0000	0	30,0000	3,7500
Slack_C4	0	1,5000	2,0000	0	0	0	1,0000	8,0000	4,0000
	C(j)-Z(j)	10,0000	15,0000	0	0	0	0	0	

Iterasi 2

Simplex Tableau -- Iteration 2

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	10,6250	0	1,0000	0	-1,8750	0	43,7500	4,1176
Slack_C2	0	1,4375	0	0	1,0000	-0,3125	0	30,6250	21,3043
X2	15,0000	0,6250	1,0000	0	0	0,1250	0	3,7500	6,0000
Slack_C4	0	0,2500	0	0	0	-0,2500	1,0000	0,5000	2,0000
	C(j)-Z(j)	0,6250	0	0	0	-1,8750	0	56,2500	

Lampiran 1. (Lanjutan)

Iterasi 3

Simplex Tableau -- Iteration 3

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	0	0	1,0000	0	8,7500	-42,5000	22,5000	
Slack_C2	0	0	0	0	1,0000	1,1250	-5,7500	27,7500	
X2	15,0000	0	1,0000	0	0	0,7500	-2,5000	2,5000	
X1	10,0000	1,0000	0	0	0	-1,0000	4,0000	2,0000	
	C(j)-Z(j)	0	0	0	0	-1,2500	-2,5000	57,5000	

Lampiran 2. Langkah Perhitungan untuk Mencari Z^1 Menggunakan Program WinQSB

Memasukkan data

	OBJ/Constraint/Bound
Maximize	10x1+15x2
C1	20x1+15x2<=110
C2	3x1+2.5x2<=44
C3	5x1+8x2<=33
C4	1.5x1+2x2<=8.8
Integer:	
Binary:	
Unrestricted:	
X1	>=0, <=M
X2	>=0, <=M

Solve and Analyze → Solve and Display Steps.

Iterasi 1

Simplex Tableau -- Iteration 1

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	20,0000	15,0000	1,0000	0	0	0	110,0000	7,3333
Slack_C2	0	3,0000	2,5000	0	1,0000	0	0	44,0000	17,6000
Slack_C3	0	5,0000	8,0000	0	0	1,0000	0	33,0000	4,1250
Slack_C4	0	1,5000	2,0000	0	0	0	1,0000	8,8000	4,4000
	C(j)-Z(j)	10,0000	15,0000	0	0	0	0	0	

Iterasi 2

Simplex Tableau -- Iteration 2

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	10,6250	0	1,0000	0	-1,8750	0	48,1250	4,5294
Slack_C2	0	1,4375	0	0	1,0000	-0,3125	0	33,6875	23,4348
X2	15,0000	0,6250	1,0000	0	0	0,1250	0	4,1250	6,6000
Slack_C4	0	0,2500	0	0	0	-0,2500	1,0000	0,5500	2,2000
	C(j)-Z(j)	0,6250	0	0	0	-1,8750	0	61,8750	

Lampiran 2. (Lanjutan)

Iterasi 3

Simplex Tableau -- Iteration 3

		X1	X2	Slack_C1	Slack_C2	Slack_C3	Slack_C4		
Basis	C(j)	10,0000	15,0000	0	0	0	0	R. H. S.	Ratio
Slack_C1	0	0	0	1,0000	0	8,7500	-42,5000	24,7500	
Slack_C2	0	0	0	0	1,0000	1,1250	-5,7500	30,5250	
X2	15,0000	0	1,0000	0	0	0,7500	-2,5000	2,7500	
X1	10,0000	1,0000	0	0	0	-1,0000	4,0000	2,2000	
	C(j)-Z(j)	0	0	0	0	-1,2500	-2,5000	63,2500	

Lampiran 3. Langkah Perhitungan Nilai λ=1-t Menggunakan Program WinQSB

Keterangan: $\lambda = x_3$

Memasukkan data

	OBJ/Constraint/Bound
Maximize	x3
C1	10x1+15x2-5.75x3>=57.5
C2	20x1+15x2+10x3<=110
C3	3x1+2.5x2+4x3<=44
C4	5x1+8x2+3x3<=33
C5	1.5x1+2x2+0.8x3<=8.8
Integer:	
Binary:	
Unrestricted:	
X1	>=0, <=M
X2	>=0, <=M
X3	>=0, <=M

Solve and Analyze → Solve and Display Steps.

Iterasi 1

		X1	X2	X3	Surplus_C1	Slack_C2	Slack_C3	Slack_C4	Slack_C5	Artificial_C1		
Basis	C(j)	0	0	1,0000	0	0	0	0	0	0	R. H. S.	Ratio
Artificial_C1	-M	10,0000	15,0000	-5,7500	-1,0000	0	0	0	0	1,0000	57,5000	3,8333
Slack_C2	0	20,0000	15,0000	10,0000	0	1,0000	0	0	0	0	110,0000	7,3333
Slack_C3	0	3,0000	2,5000	4,0000	0	0	1,0000	0	0	0	44,0000	17,6000
Slack_C4	0	5,0000	8,0000	3,0000	0	0	0	1,0000	0	0	33,0000	4,1250
Slack_C5	0	1,5000	2,0000	0,8000	0	0	0	0	1,0000	0	8,8000	4,4000
	C(j)-Z(j)	0	0	1,0000	0	0	0	0	0	0	0	
	* Big M	10,0000	15,0000	-5,7500	-1,0000	0	0	0	0	0	0	

Lampiran 3. (Lanjutan)

Iterasi 2

		X1	X2	X3	Surplus_C1	Slack_C2	Slack_C3	Slack_C4	Slack_C5	Artificial_C1		
Basis	C(j)	0	0	1,0000	0	0	0	0	0	0	R. H. S.	Ratio
X2	0	0,6667	1,0000	-0,3833	-0,0667	0	0	0	0	0,0667	3,8333	M
Slack_C2	0	10,0000	0	15,7500	1,0000	1,0000	0	0	0	-1,0000	52,5000	3,3333
Slack_C3	0	1,3333	0	4,9583	0,1667	0	1,0000	0	0	-0,1667	34,4167	6,9412
Slack_C4	0	-0,3333	0	6,0667	0,5333	0	0	1,0000	0	-0,5333	2,3333	0,3846
Slack_C5	0	0,1667	0	1,5667	0,1333	0	0	0	1,0000	-0,1333	1,1333	0,7234
	C(j)-Z(j)	0	0	1,0000	0	0	0	0	0	0	0	
	* Big M	0	0	0	0	0	0	0	0	-1,0000	0	

Iterasi 3

		X1	X2	X3	Surplus_C1	Slack_C2	Slack_C3	Slack_C4	Slack_C5	Artificial_C1		
Basis	C(j)	0	0	1,0000	0	0	0	0	0	0	R. H. S.	Ratio
X2	0	0,6456	1,0000	0,0000	-0,0330	0	0	0,0632	0	0,0330	3,9808	6,1660
Slack_C2	0	10,8654	0,0000	0,0000	-0,3846	1,0000	0	-2,5962	0	0,3846	46,4423	4,2743
Slack_C3	0	1,6058	0,0000	0,0000	-0,2692	0	1,0000	-0,8173	0	0,2692	32,5096	20,2455
X3	1,0000	-0,0549	0,0000	1,0000	0,0879	0	0	0,1648	0	-0,0879	0,3846	M
Slack_C5	0	0,2527	0	0	-0,0044	0	0	-0,2582	1,0000	0,0044	0,5308	2,1000
	C(j)-Z(j)	0,0549	0	0	-0,0879	0	0	-0,1648	0	0,0879	0,3846	
	* Big M	0	0	0	0	0	0	0	0	-1,0000	0	

Iterasi 4

		X1	X2	X3	Surplus_C1	Slack_C2	Slack_C3	Slack_C4	Slack_C5	Artificial_C1		
Basis	C(j)	0	0	1,0000	0	0	0	0	0	0	R. H. S.	Ratio
X2	0	0,0000	1,0000	0,0000	-0,0217	0	0	0,7228	-2,5543	0,0217	2,6250	
Slack_C2	0	0,0000	0,0000	0,0000	-0,1957	1,0000	0	8,5054	-42,9891	0,1957	23,6250	
Slack_C3	0	0,0000	0,0000	0,0000	-0,2413	0	1,0000	0,8234	-6,3533	0,2413	29,1375	
X3	1,0000	0,0000	0,0000	1,0000	0,0870	0	0	0,1087	0,2174	-0,0870	0,5000	
X1	0	1,0000	0,0000	0,0000	-0,0174	0	0	-1,0217	3,9565	0,0174	2,1000	
	C(j)-Z(j)	0	0	0	-0,0870	0	0	-0,1087	-0,2174	0,0870	0,5000	
	* Big M	0	0	0	0	0	0	0	0	-1,0000	0	