PENDEKATAN ALGORITMA GENETIKA DALAM MENYELESAIKAN PERMASALAHAN *FUZZY LINEAR PROGRAMMING*

S. Dewi Lestari dan Subanar

Abstract— *Fuzzy linear programming is one of the linear programming developments which able to accommodate uncertainty in the real world. Genetic algorithm approach in solving linear programming problems with fuzzy constraints has been introduced by Lin (2008) by providing a case which consists of two decision variables and three constraint functions. Other linear programming problem arise with the presence of some coefficients which are fuzzy in linear programming problems, such as the coefficient of the objective function, the coefficient of constraint functions, and right-hand side coefficients constraint functions. In this study, the problem studied is to explain the genetic algorithm approach to solve linear programming problems where the objective function coefficients and right-hand sides are fuzzy constraint functions.*

PT Dakota Furniture study case provides a linear programming formulation with a given objective function coefficients and right-hand side coefficients are fuzzy constraint functions. This study describes the use of genetic algorithm approach to solve the problem of linear programming of PT Dakota to maximize the mean income. The genetic algorithm approach is done by simulate every fuzzy number and each fuzzy numbers by distributing them on certain partition points. Then genetic algorithm is used to evaluate the value for each partition point. As a result, the Final Value represents the coefficient of fuzzy number. Fitness function is done by calculating the value of the objective function of linear programming problems. Empirical results indicated that the genetic algorithm approach can provide a very good solution by giving some limitations on each fuzzy coefficient.

Genetic algorithm approach can be extended not only to resolve the case of PT Dakota Furniture, but can also be used to solve other linear programming case with some coefficients in the objective function and constraint functions are fuzzy.

Keywords **:** *Genetic Algorithm, Fuzzy Linear Programming, Linear Programming, Two-Phase Simplex Method*

1. PENDAHULUAN

Sejauh ini, metode untuk mengkonversi *fuzzy linear programming* ke dalam matematika programming konvensional didominasi oleh penyelesaian masalah *fuzzy linear programming*. Penelitian tentang permasalahan *fuzzy linear programming* dimana koefisien fungsi kendala bersifat *fuzzy* sudah pernah dilakukan. Proses pendekatan yang dilakukan yaitu dengan mendefinisikan fungsi keanggotaan operasi aritmatika dan α-cut dalam proses perhitungannya [1]. Pengembangan permasalahan *fuzzy linear programming* yaitu terdapat suatu koefisien fungsi tujuan dan koefisien fungsi kendala yang bersifat *fuzzy* juga sudah dilakukan. Perhitungan dalam menyelesaikan permasalahan *fuzzy linear programming* juga mendefinisikan fungsi keanggotaan, operasi aritmatika dan α-cut dalam proses perhitungannya [2].

Suatu pendekatan baru menggunakan konsep pendekatan algoritma genetika yang dapat digunakan untuk menyelesaikan permasalahan *fuzzy linear programming* dimana koefisien yang bersifat *fuzzy* terdapat pada koefisien fungsi kendala. Pendekatan algoritma genetika yang diperkenalkan untuk menyelesaikan *fuzzy linear programming* dilakukan tanpa mendefinisikan fungsi keanggotaan, operasi aritmatika dan α-cut dalam proses perhitungannya. Pendekatan algoritma genetika yang diusulkan menggunakan proses evolusi biasa. Model yang dikembangkan dalam pembentukan kromosom pendekatan algoritma genetika ini dilakukan dengan cara mendistribusikan titik partisi pada bilangan fuzzy yang dilakukan untuk semua koefisien fungsi kendala yang bersifat *fuzzy* [3].

Permasalahan lain timbul dengan memberikan suatu rumusan *linear programming* dimana koefisien fungsi tujuan dan ruas kanan fungsi

Siska Dewi Lestari adalah mahasiswa Program Pascasarjana Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta (e-mail:siska_dewi_lestari@yahoo.co.id)

Subanar adalah staf pengajar Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Sekip Utara, Yogyakarta, 55281 (e-mail: subanar@yahoo.com).

kendala bersifat *fuzzy* seperti pada permasalahan yang diperkenalkan oleh Susanto [2]. Jika dalam penelitian Lin [3] permasalahan yang dipaparkannya hanya terbatas permasalahan dengan dua variabel saja dengan koefisien fungsi kendala bersifat *fuzzy*, maka dalam penelitian ini akan dikembangkan suatu permasalahan *linear programming* yang fleksibel yaitu koefisien fungsi tujuan dan fungsi kendala bersifat *fuzzy* agar bisa digunakan untuk permasalahan-permasalahan lain dari *linear programming* dengan lebih dari dua variabel keputusan. Selain itu juga pada penelitian ini akan mencoba melakukan pembentukan kromosom dengan membagi pendistribusian titik partisi pada bilangan *fuzzy* dilakukan untuk masing-masing koefisien yang bersifat *fuzzy*. Selanjutnya dalam penelitian ini akan dilakukan analisis terhadap tingkat keefektifan dan kehandalan pendekatan algoritma genetika dalam menyelesaikan permasalahan *fuzzy linear programming.*

2. Metode Penelitian

2.1. *Linear Programming* Metode Simpleks 2 Fase

Bentuk standar masalah pemprograman linear ditunjukkan pada Persamaan (1). Asumsi bahwa bentuk standar berisi n variabel (x_1, x_2, . . ., x_n) adalah sebagai berikut [4]:

Maksimumkan $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

(atau minimum)

terhadap suatu kendala

$$+ \quad + \ldots + \quad =$$
$$+ \quad + \ldots + \quad =$$
$$\cdots$$
$$+ \quad + \ldots + \quad =$$

(1)

Variabel non-negatif : $x_j \geq 0$

dimana,

z : fungsi tujuan

x_n : variabel keputusan ke – n

c_n : parameter fungsi tujuan ke – n

b_m : kapasitas kendala ke- m

a_{mn} : parameter fungsi kendala ke – m untuk variabel keputusan ke – n

Jika didefinisikan kembali

$$= \quad \begin{matrix} & & \cdots & \\ & & \cdots & \\ \vdots & \vdots & & \vdots \\ & & \cdots & \end{matrix}$$

dan

$$\mathbf{x} = \vdots \quad , \quad \mathbf{b} = \vdots$$

kendala pada Persamaan (1) dapat dituliskan kembali sistem dari persamaan ***A*x=b**.

Untuk menyelesaikan masalah LP yang memiliki variabel *artificial* dapat digunakan metode **simpleks dua fase**. Sebelum melakukan komputasi, harus dipastikan apakah fisibel solusi ada, dengan variabel *artificial* = 0. Caranya, pertama, gunakan metode simpleks untuk menyelesaikan masalah meminimalkan jumlah dari variabel *artificial*. Jika = 0, berarti ada solusi. Tetapi jika jumlahnya tidak sama dengan 0, berarti kendala tidak dapat dipenuhi. Kemudian gunakan solusi akhir sebagai solusi awal untuk masalah yang sebenarnya.

Dua fase dari metode ini adalah sebagai berikut :

Fase 1 : Susun sebuah fungsi objektif baru yang memuat jumlah dari variabel *artificial*. Gunakan metode simpleks untuk meminimalkan fungsi objektif yang memenuhi kendala.Jika fungsi objektif *artificial* dapat direduksi menjadi 0, maka setiap (non-negative) variabel *artificial* akan =0. Dalam kasus ini, semua kendala pada permasalahan awal dipenuhi, maka dapat dilanjutkan fase 2. Sebaliknya, berarti infisibel.

Fase 2 : Gunakan solusi fisibel dasar dari fase 1 (abaikan variabel *artificial*) sebagai solusi awal untuk permasalahan dengan fungsi objektif yang sebenarnya. Gunakan metode simpleks biasa untuk mendapatkan solusi optimal.

2. 2. *Fuzzy Linear Programming*

Permasalahan *linear programming* klasik adalah untuk menemukan suatu nilai dari variabel yang tidak diketahui seperti memaksimalkan fungsi tujuan pada batasan yang direpresentasikan sebagai pertidaksamaan atau persamaan linear. Permasalahan linear pada dapat dituliskan kembali sebagai :

$$\max \quad \boldsymbol{cx}$$
$$\text{dengan kendala} \quad \boldsymbol{Ax} \leq \boldsymbol{b} \qquad (2)$$
$$\boldsymbol{x} \geq 0$$

dimana $\mathbf{x} = (x_1, \ldots, x_n)^T \in R^n$ adalah variabel keputusan, $\mathbf{c} = (c_1, \ldots, c_n)$ adalah koefisien fungsi tujuan, dan $\boldsymbol{A} = [a_{ij}] \in R^{mxn}$ matrix kendala dengan elemen a_{ij} yang disebut sebagai koefisien kendala, dan $\mathbf{b} = (b_1, \ldots, b_m)$ adalah ruas kanan [5].

Pada kondisi yang sebenarnya, keputusan *linear programming* dapat disusun dalam lingkungan yang samar, sehingga sejumlah modifikasi terhadap model tersebut perlu dilakukan. Kendala mungkin bersifat samar, misalnya tanda $\leq$ mungkin merupakan batasan yang dapat sedikit dilanggar dalam batas yang dapat diterima akal sehat. Hal tersebut dapat terjadi jika kendala merepresentasikan kebutuhan sensorik atau biologic yang tidak dapat diperkirakan oleh angka yang tegas. Oleh sebab itu koefisien vektor b, atau c, atau matrik A itu sendiri dalam model *linear programming* juga mempunyai karakter yang samar karena secara alamiah tidak dapat dibatasi secara tegas.

2.2.1. Pemprograman Linear dengan Koefisien Fungsi Tujuan Berbentuk *Fuzzy*

Mengacu pada Persamaan (2), pemprograman linear dengan koefisien fungsi tujuan berupa bilangan *fuzzy* dapat ditulis sebagai berikut [5] :

$$\max \quad \boldsymbol{x}$$
$$\text{dengan kendala} \quad \boldsymbol{Ax \leq b} \qquad (3)$$
$$\boldsymbol{x \geq 0}$$

dimana $= (\tilde{\ } , \ldots , \tilde{\ })$ adalah vektor dari bilangan fuzzy.
Diasumsikan adalah bilangan fuzzy segitiga dengan fungsi keanggotaan $\tilde{\ }$; , , . Diberikan simbol $\tilde{\ }$, , , sehingga persamaan (3) menjadi [5]:

$$\max \quad (\quad , \quad , \quad)$$
$$\text{dengan kendala} \quad \boldsymbol{Ax \leq b} \qquad (4)$$
$$\boldsymbol{x \geq 0}$$

dimana $\mathbf{c}^{-} = (\ , \ldots , \ )$, $\mathbf{c}^{0} = (\ , \ldots , \ )$ dan $\mathbf{c}^{+} = (\ , \ldots , \ )$.

2.2.2. Pemprograman Linear dengan Koefisien Fungsi Kendala Berbentuk *Fuzzy*

Mengacu pada Persamaan (2), pemprograman linear dengan koefisien fungsi kendala berbentuk *fuzzy* dirumuskan sebagai berikut [5] :

$$\max \quad \boldsymbol{cx}$$
$$\text{dengan kendala} \quad \boldsymbol{x \leq b} \qquad (5)$$
$$\boldsymbol{x \geq 0}$$

dalam hal ini $= [\ ]$ adalah matriks koefisien yang bersifat *fuzzy*. Diasumsikan bahwa $= [\ ]$ adalah bilangan *fuzzy* segitiga, yang berarti adalah , , dan $= (\ , \ , \ )$, dimana $= [\ ]$, $= [\ ]$ dan $= [\ ]$. Sehingga Persamaanan (5) ditulis menjadi:

$$\max \quad \boldsymbol{cx}$$
$$\text{dengan kendala} \quad (\quad , \quad , \quad) \leq \boldsymbol{b} \qquad (6)$$
$$\boldsymbol{x \geq 0}$$

dari kedua permasalahan dasar *fuzzy linear programming* pada Persamaan (3) dan (5), tipe lainnya dari *fuzzy linear programming* adalah kombinasi dari tiga permasalahan dasar tadi dengan memberikan permasalahan dimana semua koefisiennya bilangan *fuzzy* [5]:

$$\text{Max}$$
$$\text{dengan kendala} \quad \mathbf{x} \leq \quad ,$$
$$\mathbf{x \geq 0}$$

(7)

diasumsikan $\tilde{\ }$, dan merupakan bilangan *fuzzy* segitiga, sehingga dapat dituliskan $\tilde{\ } = (\ , \ , \ )$, $= (\ , \ , \ )$ dan $= (\ , \ , \ )$.

2.3 Algoritma Genetika

Algoritma genetika sebagai cabang dari algoritma evolusi merupakan metode *adaptive* yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam atau "siapa yang kuat dia yang bertahan (*survive*)". Dengan meniru teori evolusi ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

2.3.1 Komponen-Komponen Algoritma Genetika

- Pengkodean (*Encoding*)

Pengkodean suatu solusi masalah ke dalam kromosom merupakan kunci utama untuk algoritma genetik [6]. Para pakar mengklasifikasikan pengkodean ke dalam tiga hal yaitu : pengkodean biner, pengkodean real, dan pengkodean integer (bulat).

- Inisialisasi Populasi

Inisialisasi adalah proses pembangkitan kromosom (sesuai ukuran populasi) untuk dijadikan anggota populasi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan. Setiap kromosom berisi sejumlah gen. Inisialisasi populasi dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

- Nilai *Fitness*

Pada algoritma genetika , suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai kualitasnya. Fungsi itu dikenal sebagai fungsi *fitness*. Fungsi *fitness* menghasilkan suatu nilai *fitness*.

- Seleksi

Seleksi merupakan proses pemilihan orang tua (*parent*) untuk reproduksi (biasanya didasarkan pada nilai *fitness*) [7]. Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik.

- *Crossover*

Crossover (persilangan) adalah sebuah proses yang membentuk kromosom baru dari dua kromosom induk dengan menggabungkan bagian informasi dari masing-masing kromosom [7]. *Crossover* menghasilkan kromosom baru yang disebut kromosom anak (*offspring*). *Crossover* bertujuan untuk menambah keanekaragaman string dalam satu populasi dengan penyilangan antar string yang diperoleh dari reproduksi sebelumnya. Pada *crossover* terdapat satu parameter yang sangat penting, yaitu peluang *crossover* (Pc). Pc menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi.

- Mutasi

Mutasi mengubah proses secara acak nilai dari satu atau beberapa gen dalam suatu kromosom [7]. Mutasi ini berperan untuk menggantikan suatu nilai gen yang hilang dari proses seleksi yang memungkinkan munculnya kembali nilai gen yang tidak muncul pada saat inisialisasi populasi awal.

- *Elitisme*

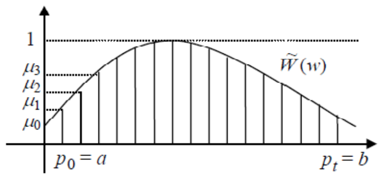
Untuk menjaga agar individu bernilai *fitness* tertinggi tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedur ini dikenal sebagai *elitism*.

2.3.2 Perumusan Model Pendekatan Algoritma Genetika untuk Menyelesaikan Permasalahan *Fuzzy Linear Programming*

Metode simpleks 2 fase dapat digunakan untuk menghasilkan solusi optimal dari permasalahan *linear programming* yang didefinisikan pada Persamaan (2) jika c_j, a_{kj} dan b_k diketahui secara pasti nilainya untuk $j = 1, 2, \ldots, n$ dan $k = 1, 2, \ldots, m$. *Fuzzy Linear Programming* adalah salah satu pengembangan dari permasalahan *linear programming* yang mampu mengakomodasi ketidakpastian yang terjadi dalam dunia nyata dimana c_j, a_{kj} dan b_k tidak dapat didekati secara pasti atau tegas (*crisp*). Oleh karena itu ketidakpastian tersebut dapat direpresentasikan menggunakan bilangan *fuzzy* dan karenanya Persamaan (2) dapat diformulasikan sebagai permasalahan *fuzzy linear programming* dengan koefisien fungsi tujuan *fuzzy* seperti pada Persamaan (3) atau permasalahan *fuzzy linear programming* dengan semua koefisiennya fuzzy seperti pada Persamaan (7).

Pendekatan algoritma genetika yang diusulkan adalah dengan menyelidiki kemungkinan dari penerapan algoritma genetika untuk menyelesaikan berbagai masalah *fuzzy linear programming* tanpa mendefinisikan fungsi keanggotaan dari bilangan *fuzzy*, tanpa menggunakan prinsip ekstensi, operasi aritmatika dan operasi α-cut untuk perhitungan *fuzzy*. Pendekatan yang ditunjukkan menggunakan proses evolusi biasa. Penggunaan algoritma genetika yang diusulkan adalah setiap bilangan *fuzzy* disimulasikan dengan mendistribusikannya pada beberapa titik partisi dan kemudian dengan menggunakan algoritma genetika untuk mengevaluasi nilai akhir pada titik partisi mewakili kelas (nilai) keanggotaan dari bilangan *fuzzy*.

Pendekatan algoritma genetika yang diperkenalkan untuk menyelesaikan permasalahan *fuzzy linear programming* adalah dengan memberikan bilangan *fuzzy triangular* $= (\; - \Delta \; , \;\; , \;\; + \Delta \;)$ yang digantikan oleh sebuah himpunan fuzzy pada interval [*a*, *b*] seperti yang diilustrasikan pada Gambar 1. di bawah ini :



Gambar 1. Himpunan fuzzy

dengan membagi interval [*a*, *b*] pada *t* partisi,

$$= \;\; + \;\; \times \; \text{——}, \; i = 0, 1, \ldots, t, \qquad (8)$$

dan p_i disebut sebagai titik partisi. Kelas keanggotaan dari pada p_i menjadi $(\;\;) =$, $i = 0, 1, \ldots, t$ dan $\mu_i \in [0,1]$. Kemudian himpunan *fuzzy* diskrit dijelaskan sebagai berikut :

$$= (\;\; , \;\; , \ldots , \;\;) = — + — + \cdots + — \qquad (9)$$

Untuk semua koefisien pada koefisien fungsi kendala dan ruas kanan fungsi kendala bersifat *fuzzy*, si pembuat keputusan dapat memberikan pelanggaran kendala. Diasumsikan bahwa adalah bilangan *fuzzy triangular* atau bilangan *fuzzy* berbentuk segitiga yang didefinisikan pada interval $[w - \Delta_1, \; w + \Delta_2]$. Interval ini dibagi menjadi *t* partisi. Diberikan $= \;\; - \Delta + \;\; \times$

$\frac{\Delta}{}$, i = 0, 1, . . ., t menjadi titik partisi dan diberikan ( ) = ∈ [0, 1], $i = 0, 1, . . ., t$, menjadi kelas (nilai) keanggotaan dari p_i pada himpunan *fuzzy* . Kemudian himpunan *fuzzy* diskrit = (, , … ,) masing-masing μ_i , $i = 0, 1, …, t$, adalah bilangan acak pada [0, 1]. Algoritma genetika digunakan untuk mengestimasi nilai w pada interval $[w - \Delta_1, w + \Delta_2]$. Setelah perhitungan *centroid* dari bilangan *fuzzy* dimana didefinisikan himpunan *fuzzy* diskrit = (, , … ,), diperoleh nilai estimasi w^* sebagai berikut :

$$^{*} = \frac{\sum \times}{\sum} \quad (10)$$

Mengikuti konsep ini, maka :

$$\sum \; ^{*} \leq \; ^{*}, \; k = 1, 2, \dots, m \quad (11)$$

a^*_{kj} adalah nilai estimasi yang dihasilkan dari perhitungan *centroid* . Sama halnya dengan b^*_k pada persamaan (3.14) juga merupakan nilai estimasi yang dijelaskan dari perhitungan centroid . Selanjutnya, *fuzzy linear programming* pada persamaan :

$$= \sum$$
$$\sum \leq , \quad = 1, 2, \dots , \quad (3.17)$$
$$\geq 0, \quad = 1, 2, \dots ,$$

menjadi :

$$= \sum$$
$$\sum \; ^{*} \leq \; ^{*}, \quad = 1, 2, \dots , \quad (3.18)$$
$$\geq 0, \quad = 1, 2, \dots ,$$

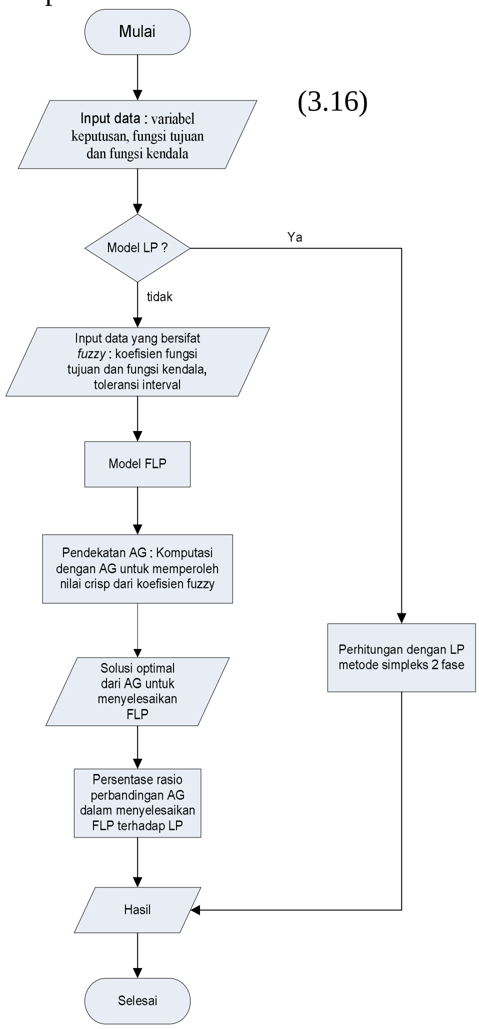
Z akan digunakan untuk menghitung fungsi *fitness* dari masing-masing kromosom [3].

Untuk permasalahan *linear programming* dimana koefisien fungsi tujuan dan koefisien fungsi kendala bersifat *fuzzy* seperti pada Persamaan (7), langkah-langkah yang digunakan untuk menyelesaikannya adalah sama. Algoritma genetika digunakan untuk mengestimasi nilai koefisien yang bersifat *fuzzy* tersebut. Dimana Z dengan koefisien hasil dari estimasi akan digunakan untuk menghitung fungsi *fitness* dari masing-masing kromosom.

2.4 Analisis dan Perancangan

Aplikasi pendekatan algoritma genetika dalam menyelesaikan permasalahan *fuzzy linear programming* merupakan suatu alternatif penyelesaian permasalahan optimasi dengan koefisien fungsi tujuan dan fungsi kendala bersifat *fuzzy*. Konsep pemodelan yang diterapkan adalah pemprograman linear *fuzzy* dengan fungsi tujuan tunggal. Model dikembangkan dalam beberapa tahapan. Pertama, mengidentifikasi variabel keputusan, fungsi kendala dan fungsi tujuan. Kedua, menetapkan parameter *fuzzy* yaitu mengidentifikasi koefisien pada fungsi tujuan dan fungsi kendala yang bersifat *fuzzy* kemudian dengan memberikan toleransi interval (Δ) atau batasan untuk masing-masing koefisien yang bersifat *fuzzy*. Ketiga, algoritma genetika digunakan untuk mengaproksimasi koefisien *fuzzy* pada fungsi tujuan dan fungsi kendala dalam permasalahan *linear programming*. Gambar 2. Menunjukkan proses optimasi dalam sistem.

(3.16)



Gambar 2. Proses optimasi dalam sistem

2.4.1 Rancangan pendekatan algoritma genetika dalam menyelesaikan permasalahan *fuzzy linear programming*

Untuk menyelesaikan permasalahan model *fuzzy linear programming*, terlebih dahulu harus menset sejumlah ukuran populasi, probabilitas *crossover* (pc), probabilitas mutasi (pm) dan maksimum generasi serta jumlah t titik partisi sebagai panjang dari masing-masing kromosom. Proses yang diperlukan dalam sistem optimasi dengan menggunakan algoritma genetika adalah sebagai berikut :

- Membangkitkan populasi awal

Membangkitkan populasi yang direpresentasikan menggunakan bilangan real. Sebuah populasi awal yang berukuran n dibangkitkan secara acak dari $[0,1]$ sesuai dengan distribusi uniform pada interval tertutup [0,1] untuk masing-masing koefisien *fuzzy*.

Pada sistem ini nantinya pembentukan kromosom dapat dilakukan dengan memilih pembagian *t* titik partisi, yaitu :

- *t* jumlah titik partisi dilakukan untuk menghitung nilai estimasi pada semua koefisien *fuzzy*.

Contohnya : *t* = 10, maka akan dibangkitkan secara acak populasi awal dari $[0,1]$

		…	
0.66	0.98	…	0.56

Sehingga panjang kromosom yang terbentuk adalah 11. Kromosom inilah yang akan digunakan untuk menghitung nilai estimasi koefisien yang bersifat *fuzzy*

- Pilihan yang kedua yaitu, *t* jumlah titik partisi dibangkitkan pada masing-masing koefisien yang bersifat *fuzzy*.

Sehingga jika terdapat 6 koefisien yang bersifat *fuzzy* dengan *t* = 10, maka akan dibangkitkan secara acak populasi awal dari $[0,1]$ sebanyak 6 *x* (10+1) = 66 gen (panjang kromosom). Gen ke-1 sampai ke-11 digunakan untuk mengestimasi koefisien *fuzzy* yang pertama, gen ke-12 sampai ke-22 digunakan untuk mengestimasi koefisien *fuzzy* yang kedua, dan seterusnya.

, , …,	, , …,	…	, , …,
0.24, 0.67, …, 0.48	0.78, 0.06, … , 0.56	…	0.66, 0.098, …, 0.72
1 11	12 22	…	56 66

dan seterusnya sebanyak ukuran populasi yang ditentukan.

- Menghitung nilai *fitness* untuk masing-masing kromosom

Untuk menghitung nilai *fitness* terlebih dahulu setiap kromosom $, h = 1,2,\dots,$ pada populasi dievaluasi, masing-masing kromosom di estimasi dengan menggunakan Persamaan (10) sehingga diperoleh koefisien yang bersifat tegas seperti pada Persamaan (11). Selanjutnya melakukan perhitungan fungsi *fitness* dengan menggunakan metode simpleks 2 fase.

Mengacu pada Persamaan (9) diberikan contoh kromosom t = 10 maka bilangan random $[0,1]^{t+1}$:

0.6641 0.9854 0.2041 0.0659 0.0143 0.3456 0.5402 0.2093 0.5167 0.8769 0.5628

jika dituliskan ke dalam himpunan *fuzzy* diskrit dengan p_i adalah titik partisi yang diberikan, $i = 0,2,\dots,t$ dan t= 10.

$$= \frac{0.6641}{} + \frac{0.9854}{} + \cdots + \frac{0.5628}{}$$

Contoh : Untuk koefisien fungsi tujuan *fuzzy* c_1 = 60, dengan toleransi intervalnya (5.0, 2.0) sehingga koefisien c_1 didefinisikan pada interval [55, 62], a= 55 dan b= 62. Sehingga mengacu pada Persamaan (8):

$$= 55 + 0 \times \frac{62 - 55}{10} = 55$$

Tabel 1. menunjukkan hasil perhitungan titik partisi untuk kromosom pertama.

Tabel 1. Perhitungan titik partisi untuk kromosom pertama dengan t=10

i	$= + \times \frac{-}{}$
0	55
1	55.7
2	56.4
3	57.1
4	57.8
5	58.5
6	59.2
7	59.9
8	60.6
9	61.3
10	62

Maka

$$^{*} = \frac{(\,.\quad \times \quad)\;(\,.\quad \times \;.\,)\;\cdots\;(\,.\quad \times \quad)}{.\qquad .\qquad \cdots\; .} = 58.6137$$

Sehingga nilai tegas untuk c_1 = 58.6137

Untuk menghitung nilai tegas pada koefisien c_2, c_3, kemudian b_1, b_2, b_3. Caranya sama, nilai keanggotaan yang digunakan juga sama, yang berbeda adalah nilai masing-masing koefisien dan toleransi intervalnya. Dari persamaan linear yang baru, kemudian lakukan perhitungan nilai fungsi tujuannya dengan menggunakan metode simpleks 2 fase. Perhitungan ini adalah nilai *fitness* dari setiap kromosomnya.

- Seleksi

Proses seleksi menggunakan metode *roulette wheel* pada n waktu.

- Pembentukan *Crossover*

Crossover adalah kunci kekuatan algoritma. Tujuan dari *crossover* adalah untuk menghasilkan suatu kelompok yang memiliki struktur kinerja yang lebih tinggi. Metode *crossover* yang digunakan di sini adalah menggunakan metode *single point* atau *one-point method* (metode satu titik), yang secara acak memilih satu titik potong dan pertukaran bagian kanan dari 2 orang tua untuk menghasilkan keturunan, dengan memberikan P_c adalah probabilitas *crossover*, $0 \leq \quad \leq 1$.

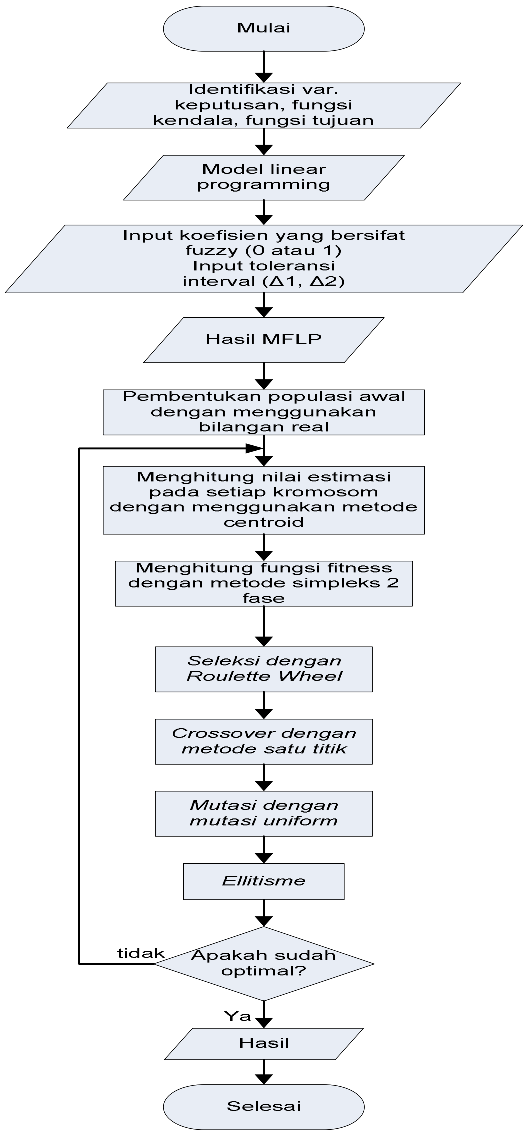
- Proses Mutasi

Mutasi adalah operator paling belakang yang menghasilkan perubahan acak dalam berbagai kromosom. Mutasi mereset satu posisi yang dipilih secara acak dari suatu bilangan real pada [0,1] terhadap tingkat mutasi. Diberikan adalah probabilitas mutasi, $0 \leq \quad \leq 1$. Biasanya adalah nilai yang sangat kecil, jika Probabilitas mutasi yang telah ditetapkan adalah 0.1 dari ukuran populasi. Maka diperoleh kromosom yang dimutasi 10% dari 10 kromosom, yaitu 1 buah kromosom.

- *Ellitisme*

Proses *ellitisme* bertujuan untuk mempertahankan kromosom dengan *fitness* tertinggi untuk tetap hidup pada generasi selanjutnya. Hasil akhir diperoleh setelah K generasi. Kromosom terbaik yang terpilih dimana memiliki fungsi *fitness* yang optimal yang merepresentasikan solusi permasalahan model *fuzzy linear programming.*

Rancangan sistem pendekatan algoritma genetika menggunakan diagram alir untuk mengetahui setiap prosesnya. Gambar 3. memperlihatkan diagram alir optimasi model *fuzzy linear programming* menggunakan pendekatan algoritma genetika.



Gambar 3. Diagram alir optimasi MFLP dengan algoritma genetika

3. Hasil Dan Pembahasan

Data yang digunakan dalam penelitian ini diambil dari data penelitian yang digunakan oleh Susanto[2]. Dimana terdapat permasalahan *linear programming* dengan koefisien fungsi tujuan bersifat *fuzzy* dan koefisien ruas kanan fungsi kendala bersifat *fuzzy*.

Maksimum

$Z = 60 \quad + 30 \quad + 20$

dengan batasan

$$8 \quad + 6 \quad + \quad \leq 48$$
$$4 \quad + 2 \quad + 1.5 \quad \leq 20$$
$$2 \quad + 1.5 \quad + 0.5 \quad \leq 8$$
$$\leq 5$$
$$x_1, x_2, x_3 \geq 0$$

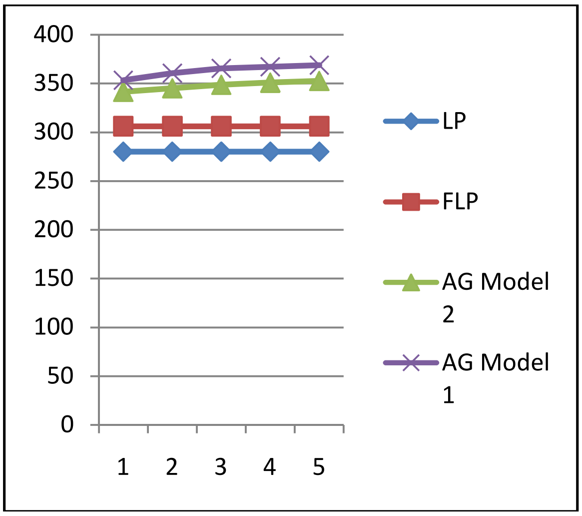
dan Tabel 2. toleransi interval (Δ) yang digunakan pada permasalahan diatas adalah :

Tabel 2. Toleransi interval (Δ) pada kasus PT. Dakota Furniture

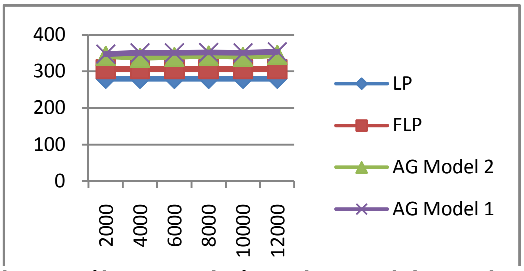
z	c_1	c_2	c_3	
	(5.0, 2.0)	(2.0, 5.0)	(3.0, 2.0)	
k	a_{k1}	a_{k2}	a_{k3}	b_k
1	(0, 0)	(0, 0)	(0, 0)	(0, 10)
2	(0, 0)	(0, 0)	(0, 0)	(0, 5)
3	(0, 0)	(0, 0)	(0, 0)	(0, 3)
4	(0, 0)	(0, 0)	(0, 0)	(0, 0)

dari toleransi interval diatas menjelaskan bahwa range dari masing-masing koefisien yang bersifat *fuzzy* adalah c_1 = [55, 62], c_2 = [28, 35] dan c_3 = [17, 22] kemudian untuk ruas kanan fungsi kendala b_1 = [48, 58], b_2 = [20, 25], b_3 = [48, 51] sedangkan toleransi interval (0, 0) menunjukkan bahwa koefisien tersebut bersifat tegas atau sudah pasti.

Untuk menguji keefektifan dan kehandalan ada dua cara yang akan ditunjukkan pada pendekatan algoritma genetika ini. Cara pertama adalah dengan memberikan jumlah generasi sama tetapi ukuran populasi yang berbeda seperti yang terlihat pada Gambar 4. Cara kedua adalah sebaliknya, yaitu dengan memberikan ukuran populasi yang sama tetapi jumlah generasi yang berbeda-beda seperti pada Gambar 5.



Gambar 4. Grafik rata-rata nilai fitness dengan 10 kali run pada ukuran populasi mulai dari 100, 200, 300, 400 dan 500 dimana AG model 1 adalah pendistribusian t untuk semua koefisien fuzzy dan AG model 2 adalah pendistribusian t untuk masing-masing koefisien fuzzy



Gambar 5. Grafik rata-rata nilai fitness dengan 10 kali run pada maksimum generasi mulai dari 2000, 4000, 6000, 8000, 10000 dan 12000 dimana AG model 1 adalah pendistribusian t untuk semua koefisien fuzzy dan AG model 2 adalah pendistribusian t untuk masing-masing koefisien fuzzy

Kemudian, dengan melakukan uji *t* (*paired two sample for means*) untuk membandingkan simpangan akurasi (% *error*) antara sistem berbasis algoritma FLP dan sistem berbasis algoritma AG dalam menyelesaikan FLP dengan menggunakan *confidance level* sebesar 95% (*alpha* = 0,05) dapat disimpulkan bahwa: (i) ada perbedaan yang signifikan pada akurasi pendeteksian antara algoritma FLP dengan algoritma hasil pendekatan AG di mana algoritma pendekatan AG lebih baik secara signifikan dari algoritma FLP, dan (ii) Pendekatan AG dalam menyelesaikan masalah FLP yang telah dilakukan ternyata berkontribusi secara signifikan dalam menghasilkam nilai Z yang lebih baik.

Untuk menentukan cut-off point keefektifan dan kehandalan dari metode algoritma genetika terhadap metode FLP pada penelitian ini dengan melihat Tabel 3. dilakukan sebagai berikut :

AG model 1 : Pendistribusian *t* untuk semua koefisien *fuzzy*

AG model 2 :Pendistribusian *t* untuk masing-masing koefisien *fuzzy*

Tabel 3. Hasil akhir rata-rata nilai Z permasalahan PT Dakota Furniture menggunakan pendekatan algoritma genetika dan FLP

	Rata-rata Z untuk Ukuran Populasi Berubah-ubah	
FLP	AG Model 1	AG Model 2
306,086	363,0102	347,7398
	Rata-rata Z untuk Maximum Generasi Berubah-ubah	
FLP	AG Model 1	AG Model 2
306,086	352,1200	340,7557

Dengan mengambil rata-rata nilai Z terbesar dikurangi dengan nilai Z FLP kemudian dibagi 2 diperoleh :

$$\frac{363{,}0102 - 306{,}086}{2} = 28{,}4621$$

Atau 306,086 + 28,4621 = 334,5481 pada nilai Z ini batas tingkat kenaikannya dapat dikatakan lebih baik dibandingkan metode FLP.

Sehingga nilai persentase dikatakan naik secara signifikan adalah :

$$\frac{28{,}4621}{306{,}086} \times 100\% = 9{,}298\%$$

terhadap metode FLP. Dengan demikian dapat disimpulkan bahwa hasil pendekatan algoritma genetika dalam menyelesaikan permasalahan FLP memberikan kontribusi secara signifikan dalam menghasilkan nilai Z yang lebih optimal.

4. KESIMPULAN

Berdasarkan pembahasan pada bab-bab sebelumnya maka diambil kesimpulan sebagai berikut :

1. Pendekatan algoritma genetika dapat digunakan untuk menyelesaikan permasalahan *fuzzy linear programming* dengan koefisien fungsi tujuan *fuzzy* dan koefisien fungsi kendala bersifat *fuzzy*.
2. Pendekatan AG pada permasalahan FLP untuk kasus PT Dakota Furniture jika pendistribusian *t* didistribusikan untuk semua koefisien *fuzzy* dengan ukuran populasi yang berubah-ubah maka nilai Z rata-ratanya adalah 363,0102sedangkan jika *t* didistribusikan untuk masing-masing koefisien *fuzzy* dengan ukuran populasi yang berubah-ubah maka nilai Z rata-ratanya adalah 347,7398. Untuk maximum generasi yang berubah-ubah dengan *t* didistribusikan untuk semua koefisien *fuzzy* nilai Z rata-ratanya adalah 352,1200 sedangkan jika *t* didistribusikan untuk masing-masing koefisien *fuzzy* nilai Z rata-ratanya adalah 340,7557. Metode FLP menghasilkan nilai Z sebesar 306,086. Metode LP dengan koefisien tegas menghasilkan nilai Z sebesar 280.
3. Keefektifan dan kehandalan pendekatan algoritma genetika dalam menyelesaikan permasalahan FLP memberikan kontribusi secara signifikan dalam menghasilkan nilai Z yang lebih optimal dibandingkan metode FLP biasa yang diperkenalkan oleh Susanto, dkk (2006) dengan penentuan *cut-off point* 9,2 %.

5. SARAN

Berdasarkan pada pengujian yang telah dilakukan pada system yang dibuat, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerjanya lebih baik, oleh karena itu saran yang diberikan adalah sebagai berikut:

1. Penelitian selanjutnya dapat dicoba perhitungan fungsi *fitness* dengan menggunakan metode lain, misalnya metode simpleks yang terevisi atau metode baru simpleks untuk mengetahui waktu komputasinya.
2. Melakukan perbandingan kehandalan dan keefektifan dengan metode lainnya (contoh metode Tabu *Search*) dalam menyelesaikan permasalahan *fuzzy linear programming*.
3. Melakukan analisis pendekatan algoritma genetika dalam menyelesaikan permasalahan *fuzzy nonlinear*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak terutama Pembimbing yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] Kumar, V.S.S., Awad S. Hanna dan P. Natarajan, 2003, Application of Fuzzy Linear Programming in Construction Projects, *International Journal of IT in Architecture, Engineering and Construction, Volume I / Issue 4/ December 2003.*

[2] Susanto, S., Dedy S., Hari A. dan YMK Aritonang, 2006, Pemodelan Pemprograman Linear dengan Koefisien Fungsi Objektif Berbentuk Bilangan Kabur Segitiga dan Kendalanya Kabur Beserta Usulan Solusinya, *Jurnal Teknik Industri Vol. 8, No. 1 Juni 2006: 14-27.* Universitas Katolik Parahyangan, Bandung.

[3] Lin, Feng-Tse, 2008, A Genetic Algorithm for Linear Programming With Fuzzy Constraints, *Journal of Information Science and Engineering 24, 801-817,* Chinese Culture University, Taipei, Taiwan.

[4] Winston, Wayne L., 2004, *Operation Research Applications and Algorithm,* Thomson Brooks/Cole, Canada.

[5] Wang, Li-Xin, 1996, *A Course in Fuzzy Systems and Control*, Prentice-Hall International, Inc., United States of America.

[6] Gen, M. dan R. Cheng, 1997, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc., United States of America.

[7] Haupt, L.R., and Haupt, E.S., 2004, *Practical Genetic Algorithm*, John Willey and Son, United States of America.