APLIKASI METODE THORANI DALAM PENYELESAIAN PERMASALAHAN PROGRAM LINEAR *FUZZY*

Mutia Dwi Haryanti, Lukman, Fitriani Agustina

Departemen Pendidikan Matematika FPMIPA UPI

Correspondent auhor: Mutiadwi03@gmail.com

ABSTRAK Program linear merupakan salah satu teknik untuk menyelesaikan persoalan pengalokasian sumber-sumber yang terbatas menggunakan persamaan dan ketidaksamaan linear dalam rangka untuk mencari pemecahan yang optimum dengan memperhatikan pembatasan-pembatasan yang ada. Permasalahan program linear memiliki parameter antara lain jumlah produk yang harus diproduksi, jumlah bahan mentah yang tersedia terbatas atau jumlah tenaga kerja yang terampil terbatas. Seringkali parameter-parameter tersebut tidak dapat diprediksi secara pasti sehingga nilainya menjadi samar (*fuzzy*). Oleh karena itu, Thorani *et al.* (2012) memperkenalkan Metode Perangkingan Thorani untuk menyelesaikan masalah pemrograman linear dengan parameternya samar (*fuzzy*). Metode ini memiliki kelebihan dibanding metode lain karena perhitungannya lebih akurat dalam membandingkan beberapa bilangan samar (*fuzzy*). Untuk membantu perhitungan, sebuah aplikasi komputer dibuat untuk memudahkan pengguna dalam memahami penyelesain masalah program linear tersebut.
Kata kunci: pemrograman linear *fuzzy*, bilangan *fuzzy*.

ABSTRACT Linear programming is one technique for solving the problems of allocating limited resources to use linear equations and inequalities in order to find the optimum solution by taking into account the existing restrictions. Parameters of linear programming problem are the amount of product to be produced, the amount of raw material available is limited or the amount of skilled labor is limited. Often these parameters can not be predicted with certainty so that its value become fuzzy. Therefore, Thorani et al. (2012) introduced a Thorani method to solve the linear programming problems where the parameters are fuzzy numbers. This method has advantages over other methods because the calculation is more accurate to compare some fuzzy numbers To help the calculation, a software designed to enable users to understand how to solve the linear programming problem.
Key words: fuzzy linear programming, fuzzy numbers.

1. Pendahuluan

Program linear (*linear programming*) merupakan salah satu metode yang berkenaan dengan pengalokasian sumber daya yang terbatas seperti, tenaga kerja terampil, bahan baku, kapasitas mesin, waktu untuk produksi, produk yang diorder dan modal dengan sebaik mungkin sehingga dicapai suatu hasil yang optimum, yaitu maksimasi atau minimasi dengan fungsi tujuan dan fungsi kendala-fungsi kendala yang diketahui. Asumsi kepastian tentang nilai-nilai parameter pada masalah pengambilan keputusan yang dimodelkan dengan program linear sering sulit dipenuhi dan tidak diketahui secara pasti. Untuk memecahkan ketidakpastian tersebut, dapat diwakili oleh bilangan *fuzzy*. Dengan adanya tingkat ketidakpastian tersebut, maka permasalahan program linear pun mengalami perkembangan menjadi program linear *fuzzy*. Secara khusus, metode yang paling sesuai untuk memecahkan masalah ini adalah didasarkan pada konsep perbandingan bilangan *fuzzy* dengan menggunakan *ranking function.*

Prinsip dasar dari metode Thorani yang terdapat dalam jurnal yang berjudul "*Ordering Generalized Trapezoidal Fuzzy Numbers*" adalah dengan mengkonversi fungsi tujuan dan fungsi kendala *fuzzy* pada masalah pemrograman linear *fuzzy* ke bentuk *crisp*. Metode ini didasarkan pada *incentre* dari *centroid* dalam perangkingan bilangan *fuzzy*. *Ranking function* didefinisikan oleh jarak Euclid antara titik pusat massa dan titik awal dalam perangkingan bilangan *fuzzy*. Sehingga permasalahan dari model program linear *fuzzy* ditransformasi ke bentuk *crisp*.

Dengan adanya perhitungan pemrograman linear *fuzzy* menggunakan aplikasi komputer (*software*) tentu akan sangat meringankan beban seseorang dalam perhitungan yang melelahkan. Hal ini bertujuan agar penyelesaian dalam permasalahan program linear *fuzzy* lebih efektif, akurat dan cepat.

Rumusan masalah dari penelitian ini adalah :

1. Bagaimana prosedur penyelesaian mencari solusi optimal dengan menggunakan metode Thorani dalam permasalahan Program Linear *Fuzzy*?
2. Bagaimana penerapan program penyelesaian permasalahan Program Linear *Fuzzy* menggunakan *Delphi 7* melalui simulasi?

Tujuan dari penelitian ini adalah :

1. Mengetahui prosedur penyelesaian mencari solusi optimal dengan menggunakan metode Thorani dalam permasalahan Program Linear *Fuzzy*.
2. Mengetahui penerapan program penyelesaian permasalahan Program Linear *Fuzzy* menggunakan *Delphi 7* melalui simulasi.

2. Kajian Pustaka

2.1 Konsep *Fuzzy*

Definisi 2.1

Misalkan U merupakan himpunan semesta. Himpunan *fuzzy* $\tilde{A}$ dari U didefinisikan oleh fungsi keanggotaan $\mu_{\tilde{A}}: U \rightarrow [0,1]$, dimana $\mu_{\tilde{A}}(x)$ adalah derajat keanggotaan x pada $\tilde{A}$, $\forall x \in U$.

Definisi 2.2

Himpunan *fuzzy* $\tilde{A}$ dari himpunan semesta U adalah normal jika dan hanya jika $\sup_{x \in U} \mu_{\tilde{A}}(x) = 1$.

Definisi 2.3

Himpunan *fuzzy* $\tilde{A}$, yang didefinisikan pada himpunan semesta dibilangan $\mathbb{R}$, dikatakan bilangan *fuzzy* jika fungsi keanggotaannya memenuhi karakteristik sebagai berikut:

i. $\mu_{\tilde{A}}: R \rightarrow [0,1]$ kontinu
ii. $\mu_{\tilde{A}}(x) = 0; \quad \forall x \in (-\infty, a] \cup [d, \infty)$
iii. $\mu_{\tilde{A}}(x)$ merupakan naik keras pada $[a, b]$ dan turun keras pada $[c, d]$
iv. $\mu_{\tilde{A}}(x) = 1; \quad \forall x \in [b, c]$

Definisi 2.3

Fungsi keanggotaan dari bilangan *fuzzy* real $\tilde{A}$ sebagai berikut :

$$f_{\tilde{A}}(x) = \begin{cases} \mu_{\tilde{A}}^{L}; & a \le x \le b \\ w; & b \le x \le c \\ \mu_{\tilde{A}}^{R}; & c \le x \le d \\ 0; & lu \end{cases}$$

dimana $0 \le w \le 1$ konstanta. $a, b, c, d \in \mathbb{R}$ dan $\mu_{\tilde{A}}^{L}: [a, b] \rightarrow [0, w]$, $\mu_{\tilde{A}}^{R}: [c, d] \rightarrow [0, w]$ adalah fungsi kontinu dan monoton keras dari $\mathbb{R}$ pada $[0, w]$

Definisi 2.7

Bilangan *fuzzy* umum $\tilde{A} = (a, b, c, d; w)$ dikatakan bilangan *fuzzy* trapesium umum jika fungsi keanggotaannya sebagai berikut :

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{w(x-a)}{(b-a)}; & a \le x \le b \\ w\,; & b \le x \le c \\ \dfrac{w(x-d)}{(c-d)}; & c \le x \le d \\ 0\,; & lu \end{cases}$$

Secara umum, jika $w = 1$, maka $\tilde{A} = (a, b, c, d; 1)$ bilangan *fuzzy* trapesium normal sedangkan jika $0 \le w \le 1$, maka $\tilde{A}$ merupakan bilangan *fuzzy* trapesium umum atau non-normal.

Jika $b = c$, maka bilangan *fuzzy* trapesium direduksi menjadi bilangan *fuzzy* segitiga yang diberikan oleh $\tilde{A} = (a, b, d; w)$. Jika w = 1, $\tilde{A} = (a, b, d)$ bilangan *fuzzy* segitiga normal sedangkan jika $0 \le w \le 1$, maka $\tilde{A}$ merupakan bilangan *fuzzy* segitiga umum atau non-normal.

2.2 Operasi Aritmatika

Misalkan diberikan dua buah bilangan *fuzzy* secara sembarang $\tilde{A}_1 = (a_1, b_1, c_1, d_1; w_1)$ dan $\tilde{A}_2 = (a_2, b_2, c_2, d_2; w_2)$ yang merupakan bilangan *fuzzy* trapesium umum, maka :

a. $\tilde{A}_1 \oplus \tilde{A}_2 = (a_1, b_1, c_1, d_1; w_1) \oplus (a_2, b_2, c_2, d_2; w_2)$
$= \left(a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2; m \quad (w_1, w_2)\right)$
dimana $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2 \in \mathbb{R}$

b. $\tilde{A}_1 \Theta \tilde{A}_2 = (a_1, b_1, c_1, d_1; w_1) \Theta (a_2, b_2, c_2, d_2; w_2)$
$= \left(a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2; m \quad (w_1, w_2)\right)$
dimana $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2 \in \mathbb{R}$

c. $\tilde{A}_1 \otimes \tilde{A}_2 = (a_1, b_1, c_1, d_1; w_1) \otimes (a_2, b_2, c_2, d_2; w_2)$
$= \left(a', b', c', d'; m \quad (w_1, w_2)\right)$
dimana $a' = m \quad (a_1 a_2, a_1 d_2, a_2 d_1, d_1 d_2)$
$b' = m \quad (b_1 b_2, b_1 c_2, c_1 b_2, c_1 c_2)$
$c' = m \quad (b_1 b_2, b_1 c_2, c_1 b_2, c_1 c_2)$
$d' = m \quad (a_1 a_2, a_1 d_2, a_2 d_1, d_1 d_2)$

d. $k\tilde{A} = (k\ , k\ , k\ , k\ ; w); k > 0$
$k\tilde{A} = (k\ , k\ , k\ , k\ ; w); k < 0$

2.3 Persoalan Pemrograman Linear *Fuzzy*

Perbedaan mendasar program linear dengan program linear *fuzzy* terletak pada representasi parameternya. Pada program linear, parameter yang ada dinyatakan sebagai bilangan tegas (*crisp*), sedangkan pada program linear *fuzzy*, parameternya direpresentasikan sebagai bilangan *fuzzy*.

Misalkan terdapat sejumlah m fungsi kendala *fuzzy* dan n variabel, memiliki ketersediaan (atau permintaan) *fuzzy* $\tilde{E}_i$ di masing-masing fungsi kendala dengan keuntungan (atau biaya) *fuzzy* $\tilde{c}_j$, dapat dirumuskan sebagai berikut:

Maksimum (atau minimum) : $\tilde{z} \approx \tilde{C}^T \otimes X$

dengan kendala : $\tilde{A}X(\tilde{\leq}, \approx, \tilde{\geq})\tilde{E}$

$$X \geq 0 \qquad (2.1)$$

dimana

$$\tilde{C}^T = \left[\tilde{c}_j\right]_{1x} \ , X = \left[x_j\right]_{n\ 1}, \tilde{A} = \left[\tilde{a}_i\right]_m \ , \tilde{E} = \left[\tilde{E}_i\right]_{m\ 1}$$

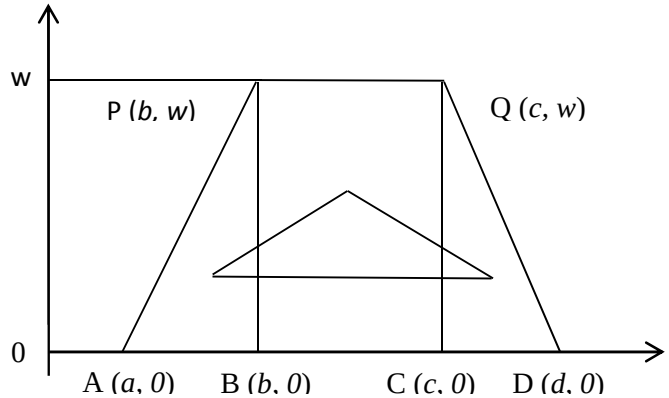
dan

$$x_j \in R, \tilde{c}_j, \tilde{a}_i \ , \tilde{E}_i \in F(R)$$

3. Metode Thorani

3.1 Metode Perangkingan Thorani

Perhatikan trapesium APQD pada Gambar 3.1. *Centroid* (titik berat atau pusat massa) dari sebuah trapesium merupakan sebuah titik keseimbangan dari trapesium tersebut. Untuk lebih jelasnya, pertama-tama yang dilakukan yaitu membagi trapesium APQD menjadi tiga bidang, yaitu sebuah bidang persegi panjang $B\ QC$ dan dua buah bidang segitiga A dan C . Misalkan titik berat dari masing-masing ketiga bidang tersebut adalah G_1, G_2, G_3. Jika ketiga titik berat tersebut dihubungkan maka akan terbentuk segitiga. Misalkan titik berat dari segitiga adalah G. G disebut *incenter* atau titik pusat lingkaran dalam segitiga $G_1G_2G_3$ yang diperoleh sebagai titik untuk mendefinisikan rangking dari bilangan *fuzzy* trapesium umum.



Gambar 3.1 Bilangan *Fuzzy* Trapesium

Berdasarkan Gambar 3.1, titik berat dari ketiga bidang tersebut pada bilangan *fuzzy* trapesium umum $\ddot{A} = (a, b, c, d; w)$ dapat dirumuskan sebagai berikut

$$T \quad b \quad A \quad : G_1 = \left(\frac{a+2b}{3}, \frac{w}{3}\right) \tag{3.1}$$

$$T \quad b \quad B \quad : G_2 = \left(\frac{b+c}{2}, \frac{w}{2}\right) \tag{3.2}$$

$$T \quad b \quad C \quad : G_3 = \left(\frac{2c+d}{3}, \frac{w}{3}\right) \tag{3.3}$$

Sedangkan, rumus *incenter* atau titik pusat lingkaran dalam segitiga $G_1G_2G_3$ dari bilangan *fuzzy* trapesium umum $\ddot{A} = (a, b, c, d; w)$ adalah

$$G = I_{\bar{A}}(\bar{x}_U, \bar{y}_U) = \left(\frac{\alpha\left(\frac{a+2b}{3}\right)+\beta\left(\frac{b+c}{2}\right)+\gamma\left(\frac{2c+d}{3}\right)}{\alpha+\beta+\gamma}, \frac{\alpha\left(\frac{w}{3}\right)+\beta\left(\frac{w}{2}\right)+\gamma\left(\frac{w}{3}\right)}{\alpha+\beta+\gamma}\right) \tag{3.4}$$

dimana

$$\alpha = \frac{\sqrt{(c-3b+2d)^2+w^2}}{6} \quad \beta = \frac{\sqrt{(2c+d-a-2b)^2}}{3} \quad \gamma = \frac{\sqrt{(3c-2a-b)^2+w^2}}{6}$$

Secara berturut-turut merupakan panjang garis $\overrightarrow{G_1G_2}, \overrightarrow{G_1G_3}, \overrightarrow{G_2G_3}$ dapat dibuktikan melalui rumus jarak euclid.

Fungsi rangking dari bilangan *fuzzy* trapesium umum $\tilde{A} = (a, b, c, d; w)$ yang memetakan himpunan semua bilangan *fuzzy* ke himpunan bilangan real adalah

$$R(\tilde{A}) = \bar{x}_U \cdot \bar{y}_U = \left(\frac{\alpha\left(\frac{a+2b}{3}\right)+\beta\left(\frac{b+c}{2}\right)+\gamma\left(\frac{2c+d}{3}\right)}{\alpha+\beta+\gamma} \cdot \frac{\alpha\left(\frac{w}{3}\right)+\beta\left(\frac{w}{2}\right)+\gamma\left(\frac{w}{3}\right)}{\alpha+\beta+\gamma} \right)$$

(3.5)

Definisi 3.1:

Mode (m) dari bilangan *fuzzy* trapesium umum $\tilde{A} = (a, b, c, d; w)$ didefinisikan :

$$m = \frac{1}{2}\int_U^w (b + c)\, d \quad = \frac{w}{2}(b + c) \tag{3.6}$$

Definisi 3.2:

Spread (s) dari bilangan *fuzzy* trapesium umum $\tilde{A} = (a, b, c, d; w)$ didefinisikan :

$$s = \int_U^w (d - a)\, d \quad = w(d - a) \tag{3.7}$$

Definis 3.3:

Left spread (l_s) dari bilangan *fuzzy* trapesium umum $\tilde{A} = (a, b, c, d; w)$ didefinisikan :

$$l_s = \int_U^w (b - a)\, d \quad = w(b - a) \tag{3.8}$$

Definisi 3.4:

Right spread (r_s) dari bilangan *fuzzy* trapesium umum $\tilde{A} = (a, b, c, d; w)$ didefinisikan :

$$r_s = \int_U^w (d - c)\, d \quad = w(d - c) \tag{3.9}$$

Dengan menggunakan definisi-definisi tersebut, selanjutnya dapat mendefinisikan prosedur perangkingan dari dua bilangan *fuzzy* trapesium umum. Misalkan $\tilde{A} = (a_1, b_1, c_1, d_1; w_1)$ dan $\tilde{B} = (a_2, b_2, c_2, d_2; w_2)$ merupakan dua bilangan *fuzzy* trapesium umum. Berikut langkah-langkah membandingkan $\tilde{A}$ dan $\tilde{B}$ berdasarkan algoritma urutan, yaitu :

Langkah 1 : Hitung nilai $R(\vec{A})$ dan $R(\vec{B})$
Kasus (i) : Jika $R(\vec{A}) > R(\vec{B})$ maka $\vec{A} > \vec{B}$
Kasus (ii) : Jika $R(\vec{A}) < R(\vec{B})$ maka $\vec{A} < \vec{B}$
Kasus (ii) : Jika $R(\vec{A}) = R(\vec{B})$ maka perbandingan tidak mungkin, lanjutkan ke langkah 2
Langkah 2 : Hitung nilai $m(\vec{A})$ dan $m(\vec{B})$
Kasus (i) : Jika $m(\vec{A}) > m(\vec{B})$ maka $\vec{A} > \vec{B}$
Kasus (ii) : Jika $m(\vec{A}) < m(\vec{B})$ maka $\vec{A} < \vec{B}$
Kasus (ii) : Jika $m(\vec{A}) = m(\vec{B})$ maka perbandingan tidak mungkin, lanjutkan ke langkah 3
Langkah 3 : Hitung nilai $s(\vec{A})$ dan $s(\vec{B})$
Kasus (i) : Jika $s(\vec{A}) > s(\vec{B})$ maka $\vec{A} > \vec{B}$
Kasus (ii) : Jika $s(\vec{A}) < s(\vec{B})$ maka $\vec{A} < \vec{B}$
Kasus (ii) : Jika $s(\vec{A}) = s(\vec{B})$ maka perbandingan tidak mungkin, lanjutkan ke langkah 4
Langkah 4 : Hitung nilai $l_2(\vec{A})$ dan $l_2(\vec{B})$
Kasus (i) : Jika $l_2(\vec{A}) > l_2(\vec{B})$ maka $\vec{A} > \vec{B}$
Kasus (ii) : Jika $l_2(\vec{A}) < l_2(\vec{B})$ maka $\vec{A} < \vec{B}$
Kasus (ii) : Jika $l_2(\vec{A}) = l_2(\vec{B})$ maka perbandingan tidak mungkin, lanjutkan ke langkah 5
Langkah 5 : Uji nilai w_1 dan w_2
Kasus (i) : Jika $w_1 > w_2$ maka $\vec{A} > \vec{B}$
Kasus (ii) : Jika $w_1 < w_2$ maka $\vec{A} < \vec{B}$
Kasus (ii) : Jika $w_1 = w_2$ maka $\vec{A} \approx \vec{B}$

Keunggulan dari metode Thorani ini adalah memiliki metode perangkingan bilangan *fuzzy* yang sederhana dan mudah dalam perhitungan, tidak hanya memberikan hasil yang lebih akurat dalam mendefinisikan masalah tetapi juga memberikan urutan perangkingan yang baik saat membandingkan beberapa bilangan *fuzzy* dibandingkan dengan metode lain.

3.2 Prosedur Penyelesaian Persoalan Pemrograman Linear *Fuzzy*

1. Membangun model pemrograman linear *fuzzy* yang diperoleh.

2. Mengkonversikan model pemrograman linear *fuzzy* ke model pemrograman linear *crisp* menggunakan fungsi perangkingan metode Thorani dengan memperhatikan algoritma urutan sebagai penentu bilangan *crisp* yang akan digunakan pada model pemrograman linear *crisp*.
   i). Jika nilai $R(\vec{A}) = R(\vec{B})$ maka gunakan nilai $m(\vec{A})$ dan $m(\vec{B})$
   ii). Jika nilai $m(\vec{A}) = m(\vec{B})$ maka gunakan nilai $s(\vec{A})$ dan $s(\vec{B})$
   iii). Jika nilai $s(\vec{A}) = s(\vec{B})$ maka gunakan nilai $l_i(\vec{A})$ dan $l_i(\vec{B})$
   iv). Jika nilai $l_i(\vec{A}) = l_i(\vec{B})$ maka gunakan nilai $w(\vec{A})$ dan $w(\vec{B})$
3. Selesaikan persoalan program linear *crisp* untuk memperoleh solusi optimal (X_j).
4. Temukan solusi optimal *fuzzy* dengan mensubstitusi nilai variabel keputusan yang sudah diperoleh ke dalam fungsi tujuan *fuzzy* $(\tilde{z} \approx \vec{C}^T \otimes X)$ pada model pemrograman linear *fuzzy*.

4. Simulasi

4.1 Pendahuluan

Misalkan terdapat suatu persoalan program linear *fuzzy*, yaitu sebuah perusahaan memproduksi 2 produk, yaitu P_1 dan P_2. Produk-produk tersebut diproses dengan 2 mesin yang berbeda yaitu M_1 dan M_2. Perincian untuk produk P_1 dan P_2 adalah sebagai berikut :

a. Waktu yang dibutuhkan mesin M_1 untuk memproduksi produk P_1 dan P_2 ditunjukkan dengan bilangan *fuzzy* trapesium, masing-masing adalah $(1,3,5,6; 0,5)$ jam dan $(1,5,7,9; 0,4)$ jam. Waktu yang dibutuhkan mesin M_2 untuk memproduksi produk P_1 dan P_2 ditunjukkan dengan bilangan *fuzzy* trapesium, masing-masing adalah $(1,2,4,7; 0,6)$ jam dan $(1,2,5,9; 0,5)$ jam.
b. Keuntungan per unit produk P_1 dan P_2 ditunjukkan dengan bilangan *fuzzy* trapesium, masing-masing adalah $(3,5,8,13; 0,6)$ juta dan $(4,6,10,16; 0,5)$ juta.
c. Jumlah kapasitas waktu untuk mesin M_1 dan M_2 ditunjukkan dengan bilangan *fuzzy* trapesium, masing-masing adalah $(2,5,8,18; 0,9)$ jam dan $(2,4,6,8; 0,7)$ jam.

4.2 Penyelesaian Persoalan Linear *Fuzzy*

4.2.1 Kasus Maksimasi

- **Langkah 1**

  Membangun model pemrograman linear *fuzzy* untuk persoalan program linear *fuzzy* di atas adalah sebagai berikut :

  M

  $$\tilde{z} \approx (3,5,8,13;0,6)\otimes x_1 \oplus (4,6,10,16;0,5)\otimes x_2$$

  $d \quad k$:

  $$(1,3,5,6;0,5)\otimes x_1 \oplus (1,5,7,9;0,4)\otimes x_2 \tilde{\leq} (2,5,8,18;0,9)$$
  $$(1,2,4,7;0,6)\otimes x_1 \oplus (1,2,5,9;0,5)\otimes x_2 \tilde{\leq} (2,4,6,8;0,7)$$
  $$x_1, x_2 \geq 0$$

- **Langkah 2**

  Mengecek semua perangkingan bilangan *fuzzy* pada model pemrograman linear *fuzzy* yang telah dibangun berdasarkan algoritma urutan, diperoleh :

  $R(3,5,8,13;0,6) = 1,625;\ R(4,6,10,16;0,5) = 1,667;$
  $R(1,3,5,6;0,5) = 0,833;\ R(1,5,7,9;0,4) = 1;$
  $R(2,5,8,18;0,9) = 2,438; R(1,2,4,7;0,6) = 0,750;$
  $R(1,2,5,9;0,5) = 0,729; R(2,4,6,8;0,7) = 1,458$

  Karena semua nilai perangkingan (R) masing-masing bilangan *fuzzy* pada model pemrograman linear *fuzzy* tidak terdapat nilai perangkingan yang sama, maka nilai perangkingan (R) dapat digunakan sebagai bilangan *crisp* dari bilangan *fuzzy* tersebut yang akan digunakan pada model pemrograman linear *crisp*.

- **Langkah 3**

  Mengkonversikan ke model pemrograman linear *crisp*, sehingga menjadi sebagai berikut :

  $M \qquad z = 1,625x_1 + 1,667x_2$

  $d \quad k \qquad : 0,833x_1 + x_2 \leq 2,438$

  $$0,750x_1 + 0,729x_2 \leq 1,458$$
  $$x_1, x_2 \geq 0$$

- **Langkah 4**

  Membangun bentuk standar model pemrograman linear *crisp*. Bentuk standar model pemrograman linear *crisp* menjadi sebagai berikut :

$$M \qquad z - 1{,}625x_1 - 1{,}667x_2 - 0x_3 - 0x_4$$
$$d \quad k \quad : 0{,}833x_1 + x_2 + x_3 = 2{,}438$$
$$0{,}750x_1 + 0{,}729x_2 + x_4 = 1{,}458$$
$$x_1, x_2, x_3, x_4 \geq 0$$

- **Langkah 5**

Menguji keoptimuman penyelesaian program linear *crisp* dengan menggunakan metode simpleks seperti yang disajikan pada Tabel 4.3.

Tabel 4.3 Solusi Awal Perankingan Thorani

X_b	x_1	x_2	x_3	x_4	**Indeks**
$x_3 = 2{,}438$	0,833	1	1	0	2,438
$x_4 = 1{,}458$	0,750	0,729	0	1	2
$z = 0$	$-1{,}625$	$-1{,}667$	0	0	

Perhatikan Tabel 4.3, ternyata pemecahan ini belum optimal, sebab masih terdapat nilai $z_j - c_j < 0$. Dua dari $(z_j - c_j)$ negatif yaitu $-1{,}625$ dan $-1{,}667$, karena $(z_j - c_j)$ dari kolom 2 nilainya paling negatif, maka dari itu x_2 masuk ke basis dalam Tabel 4.4 (tabel baru).

Tabel 4.4 Solusi Akhir

X_b	x_1	x_2	x_3	x_4
$x_3 = 0{,}438$	$-0{,}196$	0	1	$-1{,}372$
$x_2 = 2$	1,029	1	0	1,372
$z = 3{,}334$	0,090	0	0	2,287

Karena semua $z_j - c_j \geq 0$ untuk setiap x_j, maka Tabel 4.4 sudah memberikan pemecahan yang optimal. x_2 dan x_3 dalam basis. Jadi pemecahan optimal diperoleh dengan $x_1 = 0$, $x_2 = 2$, dan nilai maksimum dari $z = 3{,}334$.

- **Langkah 6**

Mensubstitusi nilai variabel keputusan yang sudah diperoleh ke dalam fungsi tujuan *fuzzy* pada model pemrograman linear *fuzzy* untuk memperoleh solusi optimal *fuzzy*, maka :

$$M \qquad \tilde{z} \approx (3{,}5{,}8{,}13; 0{,}6) \otimes x_1 \oplus (4{,}6{,}10{,}16; 0{,}5) \otimes x_2$$
$$\approx (3{,}5{,}8{,}13; 0{,}6) \otimes 0 \oplus (4{,}6{,}10{,}16; 0{,}5) \otimes 2$$
$$\approx (8{,}12{,}20{,}32; 0{,}5)$$

Dari hasil di atas diperoleh keuntungan *fuzzy* per unit dalam memproduksi produk P_1 dan P_2 sebesar $(8,12,20,32;0,5)$. Dengan kata lain, keuntungan minimum sebesar 8 juta, maksimum 32 juta, rata-rata keuntungan antara 12 dan 20 juta, dengan derajat kepuasannya sebesar 0,5.

4.2.2 Kasus Minimasi

- **Langkah 1**

  Membangun model pemrograman linear *fuzzy* berdasarkan persoalan program linear *fuzzy* yang dikemukakan. Model pemrograman linear *fuzzy* pada kasus minimasi untuk persoalan program linear *fuzzy* di atas adalah sebagai berikut :

  M

  $$\tilde{z} \approx (-13,-8,-5,-3;0,6)\otimes x_1 \oplus (-16,-10,-6,-4;0,5)\otimes x_2$$

  d k :

  $$(1,3,5,6;0,5)\otimes x_1 \oplus (1,5,7,9;0,4)\otimes x_2 \tilde{\leq} (2,5,8,18;0,9)$$
  $$(1,2,4,7;0,6)\otimes x_1 \oplus (1,2,5,9;0,5)\otimes x_2 \tilde{\leq} (2,4,6,8;0,7)$$
  $$x_1, x_2 \geq 0$$

- **Langkah 2**

  Mengecek semua perangkingan bilangan *fuzzy* pada model pemrograman linear *fuzzy* yang telah dibangun berdasarkan algoritma urutan, diiperoleh :

  $R(-13,-8,-5,-3;0,6) = -1,625$; $R(-16,-10,-6,-4;0,5) = -1,667$;
  $R(1,3,5,6;0,5) = 0,833$; $R(1,5,7,9;0,4) = 1$;
  $R(2,5,8,18;0,9) = 2,438$; $R(1,2,4,7;0,6) = 0,750$;
  $R(1,2,5,9;0,5) = 0,729$; $R(2,4,6,8;0,7) = 1,458$

  Karena semua nilai perangkingan (R) masing-masing bilangan *fuzzy* pada model pemrograman linear *fuzzy* tidak terdapat nilai perangkingan yang sama, maka nilai perangkingan (R) dapat digunakan sebagai bilangan *crisp* dari bilangan *fuzzy* tersebut yang akan digunakan pada model pemrograman linear *crisp*.

- **Langkah 3**

  Mengkonversikan ke model pemrograman linear *crisp*, sehingga menjadi sebagai berikut :

  M $z = (-1,625)x_1 + (-1,667)x_2$

  d k : $0,833x_1 + x_2 \leq 2,438$

  $0,750x_1 + 0,729x_2 \leq 1,458$

$$x_1, x_2 \geq 0$$

- **Langkah 4**

  Membangun bentuk standar model pemrograman linear *crisp*. Bentuk standar model pemrograman linear *crisp* menjadi sebagai berikut :

$$M \qquad z + 1{,}625x_1 + 1{,}667x_2 - 0x_3 - 0x_4$$
$$d \quad k \qquad : 0{,}833x_1 + x_2 + x_3 = 2{,}438$$
$$0{,}750x_1 + 0{,}729x_2 + x_4 = 1{,}458$$
$$x_1, x_2, x_3, x_4 \geq 0$$

- **Langkah 5**

  Menguji keoptimuman penyelesaian program linear *crisp* dengan menggunakan metode simpleks seperti yang disajikan pada Tabel 4.6.

Tabel 4.6 Solusi Awal Perankingan Thorani

X_b	x_1	x_2	x_3	x_4	Indeks
$x_3 = 2{,}438$	0,833	1	1	0	2,438
$x_4 = 1{,}458$	0,750	0,729	0	1	2
$z = 0$	1,625	1,667	0	0	

Perhatikan Tabel 4.6, ternyata pemecahan ini belum optimal, sebab masih terdapat nilai $z_j - c_j > 0$. Dua dari $(z_j - c_j)$ positif yaitu 1,625 dan 1,667, karena $(z_j - c_j)$ dari kolom 2 nilainya paling positif, maka dari itu x_2 masuk ke basis dalam Tabel 4.7 (tabel baru).

Tabel 4.7 Solusi Akhir

X_b	x_1	x_2	x_3	x_4
$x_3 = 0{,}438$	$-0{,}196$	0	1	$-1{,}372$
$x_2 = 2$	1,029	1	0	1,372
$z = -3{,}334$	$-0{,}090$	0	0	$-2{,}287$

Karena semua $z_j - c_j \leq 0$ untuk setiap x_j, maka Tabel 4.7 sudah memberikan pemecahan yang optimal. x_2 dan x_3 dalam basis. Jadi pemecahan optimal diperoleh dengan $x_1 = 0$, $x_2 = 2$, dan nilai minimum dari $z = -3{,}334$.

- **Langkah 6**

  Mensubstitusi nilai variabel keputusan yang sudah diperoleh ke dalam fungsi tujuan *fuzzy* pada model pemrograman linear *fuzzy* untuk memperoleh solusi optimal *fuzzy*, maka :

  M

  $$\tilde{z} \approx (-13,-8,-5,-3;0,6)\otimes x_1 \oplus (-16,-10,-6,-4;0,5)\otimes x_2$$
  $$\approx (-13,-8,-5,-3;0,6)\otimes 0 \oplus (-16,-10,-6,-4;0,5)\otimes 2$$
  $$\approx (-32,-20,-12,-8;0,5)$$

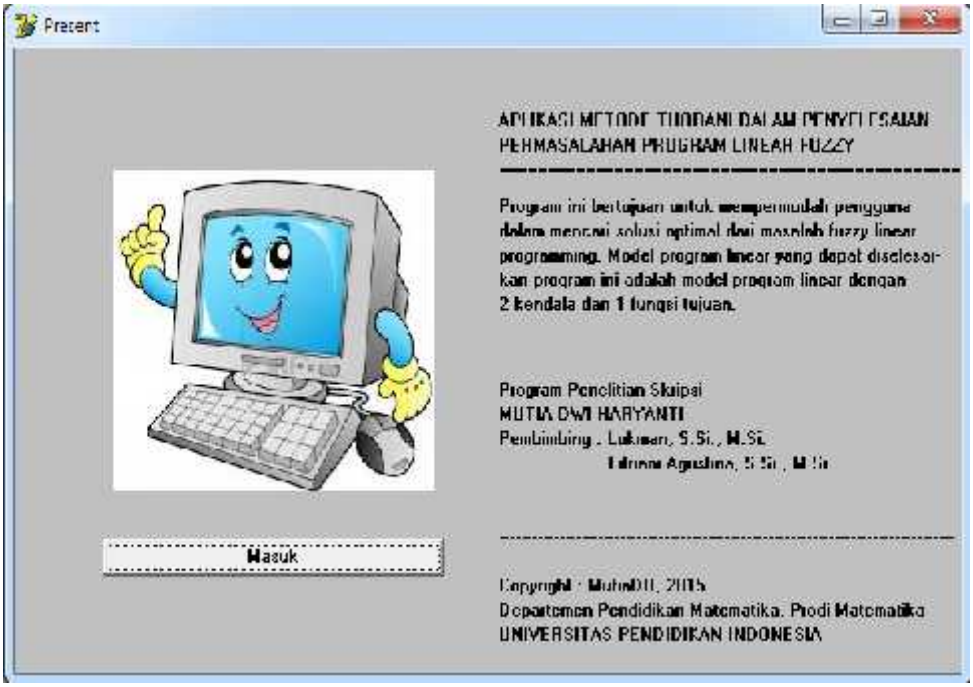
  Dari hasil di atas diperoleh keuntungan *fuzzy* per unit dalam memproduksi produk P_1 dan P_2 sebesar $(-32,-20,-12,-8;0,5)$. Dengan kata lain, keuntungan minimum sebesar 8 juta, maksimum 32 juta, rata-rata keuntungan antara 12 dan 20 juta, dengan derajat kepuasannya sebesar 0,5.

4.3 Implementasi Program

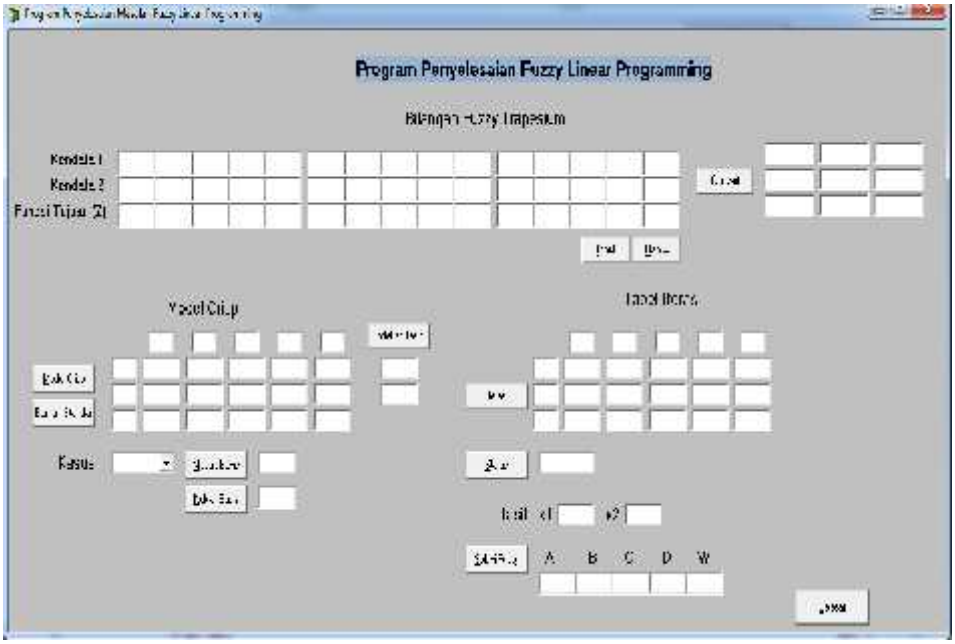
Implementasi program rancangan desain antarmuka untuk program aplikasi 'Penyelesaian Masalah *Fuzzy Linear Programming*' ini menggunakan *Borland Delphi 7*.

4.3.1 Implementasi Antarmuka

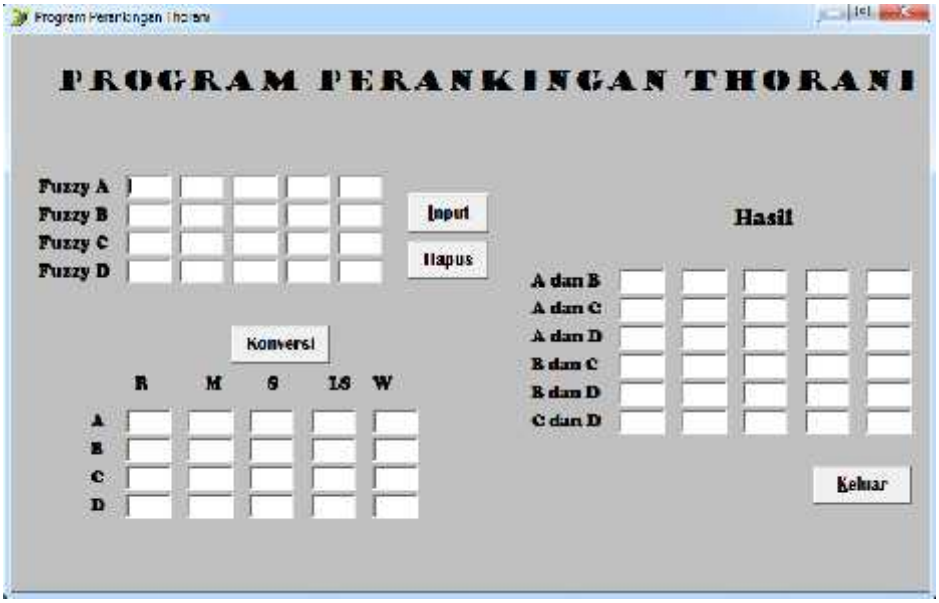
Berikut tampilan antarmuka program aplikasi yang dibuat menggunakan *Borland Delphi 7*.



Gambar 4.1 Tampilan Jendela Masuk Program



Gambar 4.2 Tampilan Jendela Program Penyelesaian *Fuzzy Linear Programming*

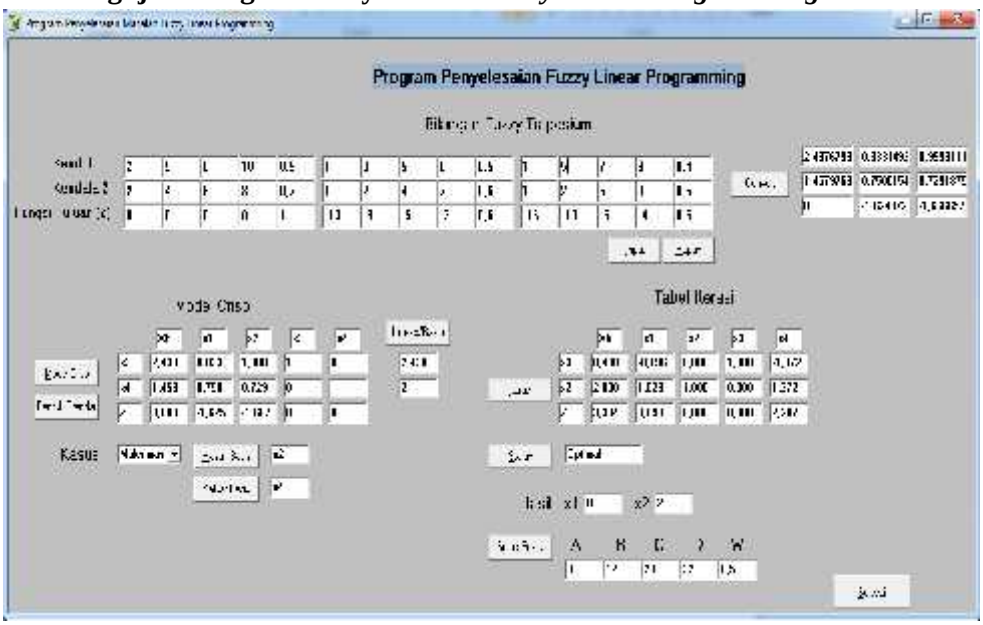


Gambar 4.3 Tampilan Jendela Program Perankingan Thorani

4.4 Pengujian Program

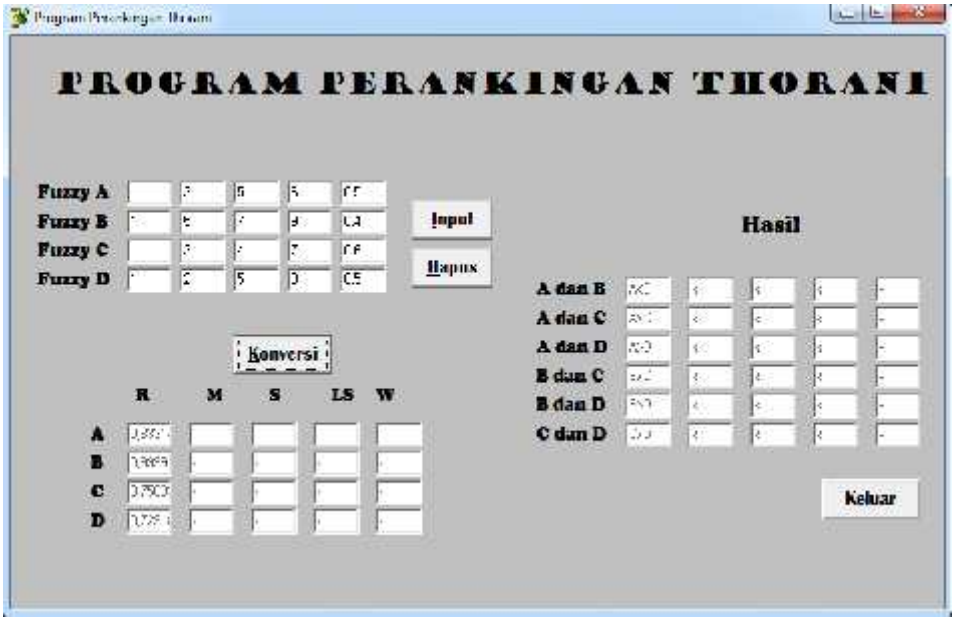
Pengujian program bertujuan agar program dapat berjalan dengan baik tanpa adanya *error*, dan sangat memungkinkan untuk dilakukan pengembangan program lebih lanjut.

4.4.1 Pengujian Program Penyelesaian *Fuzzy Linear Programming*



Gambar 4.4 Hasil Akhir Program Penyelesaian *Fuzzy Linear Programming*

4.4.2 Pengujian Program Penyelesaian *Fuzzy Linear Programming*



Gambar 4.5 Hasil Akhir Program Perangkingan Thorani

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pembahasan sebelumnya, dapat diambil kesimpulan sebagai berikut :

1. Prosedur mencari solusi optimal masalah program linear *fuzzy* menggunakan metode Thorani antara lain mengkonversi model pemrograman linear *fuzzy* ke model pemrograman linear *crisp* dengan memanfaatkan fungsi ranking dan memperhatikan algoritma urutan, lalu menyelesaikan model *crisp* yang diperoleh dengan menggunakan metode penyelesaian masalah program linear yang sudah ada, dalam penelitian ini diselesaikan dengan menggunakan metode simpleks. Perhitungan menggunakan metode Thorani ini sangat menguntungkan karena perhitungannya yang lebih akurat dalam membandingkan beberapa bilangan *fuzzy* dibandingkan dengan metode lain.
2. Program aplikasi yang dibuat dapat berjalan dengan baik dan apa yang diharapkan dari pembuatan program tersebut, yaitu kecepatan dan keakuratan dapat tercapai. Hal ini terbukti dengan membandingkan hasil penyelesaian

permasalahan program linear *fuzzy* yang dikerjakan secara manual dan dikerjakan menggunakan program aplikasi adalah sama.

5.2 Saran

1. Algoritma pemrograman dalam mencari solusi dan model *crisp* agar lebih dipermudah lagi supaya lebih efisien.
2. Dapat dikembangkan kembali program untuk penyelesaian masalah program linear *fuzzy* dengan lebih dari 2 kendala, 2 variabel dan tidak hanya untuk seluruh fungsi kendala dengan pembatas "≤". Selain menggunakan metode simpleks dalam mencari solusi optimal, seperti metode Big-M dan metode 2 fase dapat digunakan sebagai solusi pembanding.
3. Dapat dilakukan penerapan metode Thorani pada masalah program linear *fuzzy* yang terjadi di kehidupan sehari-hari, menggunakan data lapangan dari studi kasus yang lain.

REFERENSI

Bustani, H. (2005). *Fundamental Operation Research*. Jakarta: Gramedia Pustaka Utama.

Dimyati, T.T. dan Ahmad, D. (1992). *Operation Research : Model-model Pengambilan Keputusan*. Bandung: CV. Sinar Baru Bandung.

Kumar, A., P. Singh., dan J. Kaur. (2010). *Generalized Simplex Algorithm to Solve Fuzzy Linear Programming Problems with Ranking of Generalized Fuzzy Numbers*. Turkish Journal of Fuzzy Systems, 1 (2), hlm. 80-103.

Kusumadewi, S. dan Hari, P. (2010). *Aplikasi Logika Fuzzy untuk Pendukung Keputusan.* Yogyakarta: Graha Ilmu.

Munir, R. (2011). *Algoritma & Pemrograman dalam Bahasa Pascal dan C Edisi Revisi*. Bandung: Informatika.

Prawitasari, E.R. (2014). *Program Aplikasi Penyelesaian Masalah Fuzzy Transshipment Menggunakan Metode Mehar.* (Skripsi). Fakultas Pendidikan

Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Bandung.

Purcell, E. J., Dale, V., dan Steven E. R. (2003). *Kalkulus Edisi Kedelapan*. Jakarta: Erlangga.

Supranto, J. (2013). *Riset Operasi untuk Pengambilan Keputusan Edisi Ketiga*. Jakarta: PT. Raja Grafindo Persada.

Thorani, Y. L. P., P. Phani B. R., dan N. Ravi S. (2012). *Ordering Generalized Trapezoidal Fuzzy Numbers.* International Journal of Contemporary Mathematical Sciences, 7 (12), hlm. 555-573.