OPTIMASI MASALAH PROGRAM LINIER *FUZZY* TIDAK PENUH MENGGUNAKAN METODE SIMPLEKS DAN ALGORITMA DETERMINAN MATRIKS ORDO DUA

Rahmi Wahida[1], Drs. Solichin Zaki, M.Kom[2],

Drs. Kartono, M.Si[3] Program Studi Matematika FSM Universitas Diponegoro

Jl. Prof. H. Soedarto, S.H. Tembalang Semarang

rahmiwahida14@gmail.com, zaki.solichin@gmail.com

ABSTRACT. Not fully fuzzy linear programming where there is trapezoidal fuzzy number form on decision variables, objective function coefficients, constraint functions coefficients, or the right side of constraints is part of fuzzy linear programming. This essay discusses about how to determine the optimal solution of not fully fuzzy linear programming problems. The method used is Simplex Method and Determinant's Algorithm of Two Square Matrix and Robust Ranking to convert the fuzzy linear programming problems into crisp linear programming problem. Determinant of two order matrix Algorithm involves determinant of matrix in each iteration steps to obtain an optimal solution that is applicable to the maximum or minimum case. This algorithm is more efficient than usual simplex method which involves row transformations.

Keywords : Fuzzy Linear Programming, Trapezoidal Fuzzy, Robust Ranking, Simplex, Determinant of matrix.

I. PENDAHULUAN

Permasalahan dalam kehidupan nyata erat hubungannya dengan permasalahan yang mengandung ketidakpastian. Salah satu contohnya pada masalah pengalokasian sumber daya yang terbatas sehingga laba yang didapatkan oleh suatu perusahaan tersebut tidaklah pasti. Penggambaran keadaan dunia nyata yang tidak pasti inilah muncul istilah *fuzzy*. Teori himpunan *fuzzy* banyak diterapkan dalam berbagai disiplin ilmu seperti dalam pemrograman linear**.**Program linier *fuzzy* merupakan masalah program linier dengan bilangan yang digunakan adalah bilangan yang tidak pasti atau kabur. Pemrograman linear *fuzzy* digunakan untuk mencari solusi yang optimal seperti memaksimumkan keuntungan atau meminimumkan biaya produksi berdasarkan kendala dan kriteria yang dinyatakan dalam fungsi tujuan. Program linear *fuzzy* memiliki bentuk yaitu program linier *fuzzy* penuh dan program linier *fuzzy* tidak penuh.

Salah satu metode yang digunakan untuk menyelesaikan persoalan pemrograman linear *fuzzy* adalah metode simpleks. Jervin Zen Lobo mengusulkan sebuah algoritma dalam melakukan iterasi pada metode simpleks yang menggunakan determinan matriks

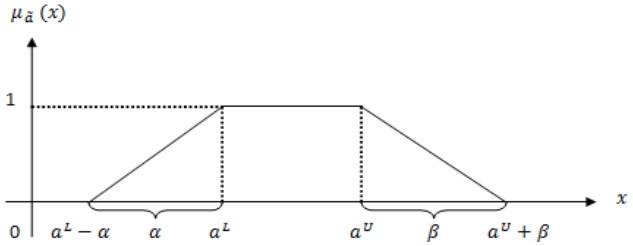
ordo dua di setiap langkah. Untuk melakukan iterasi berikutnya, tergantung pada posisi pivot dan elemen yang akan diubah untuk mendapatkan solusi optimal**.** Tulisan ini membahas tentang optimasi masalah program linier *fuzzy* tidak penuh dengan bilangan *trapezoidal fuzzy* menggunakan metode simpleks dan algoritma determinan matriks ordo dua. Metode penegasan yang digunakan adalah *Robust Rangking*.

II. HASIL DAN PEMBAHASAN

2.1. Bilangan *Trapezoidal Fuzzy*

Definisi 2.1[1] Suatu bilangan *fuzzy* $\tilde{a} = (a^l, a^u, \alpha, \beta)$, dengan $a^l, a^u, \alpha, \beta \in \mathbb{R}$ dikatakan bilangan *trapezoidal fuzzy* apabila fungsi keanggotaannya didefinisikan sebagai berikut :

$$\mu_{\tilde{a}}(x) = \begin{cases} 1 - \dfrac{a^l - x}{\alpha}, & a^l - \alpha \le x < a^l \\ 1, & a^l \le x \le a^u \\ 1 - \dfrac{x - a^u}{\beta}, & a^u < x \le a^u + \beta \\ 0, & lainnya. \end{cases}$$



Gambar 2.1 Fungsi Keanggotaan bilangan *trapezoidal fuzzy*

Definisi 2.2[2] Suatu bilangan *trapezoidal fuzzy* $\tilde{a} = (a^l, a^u, \alpha, \beta)$, dengan $a^l, a^u, \alpha, \beta \in \mathbb{R}$ dikatakan bilangan *trapezoidal fuzzy non-negative* apabila $a^l - \alpha \ge 0$.

Definisi 2.3[1] Bilangan *trapezoidal fuzzy* dapat dinyatakan dengan $\tilde{a} = (a^l, a^u, \alpha, \beta)$ dengan $a^l < a^u (a^l \ne a^u), \alpha > 0, \beta > 0$ dan $a^l, a^u, \alpha, \beta \in \mathbb{R}$. *Core* dari bilangan *trapezoidal fuzzy* $\tilde{a}$ adalah $[a^l, a^u]$ dan *support* dari bilangan *trapezoidal fuzzy*$\tilde{a}$ adalah $(a^l - \alpha, a^u + \beta)$.

2.2. Penegasan Bilangan *Trapezoidal Fuzzy*

Definisi 2.5[3] Diberikan sebuah bilangan *trapezoidal fuzzy* $\tilde{a} = (a^l, a^u, \alpha, \beta)$ dengan $a^l, a^u, \alpha, \beta \in \mathbb{R}$. *Robust Ranking* dari bilangan *trapezoidal fuzzy*$\tilde{a}$ didefinisikan sebagai berikut : $\Re(\tilde{a}) = \int_0^1 0.5(a_\lambda^l, a_\lambda^u) d\lambda$

dimana $(a_\lambda^l, a_\lambda^u)$ merupakan tingkat potongan-λ, dengan $(a_\lambda^l, a_\lambda^u) = a_\lambda^l + a_\lambda^u$, $a_\lambda^l = \alpha\lambda + (a^l - \alpha)$, dan $a_\lambda^u = (a^u + \beta) - \beta\lambda$. Nilai *Robust Ranking* $\Re(\tilde{a})$ merupakan nilai perwakilan dari bilangan *fuzzy* $\tilde{a}$.

Lemma 2.1[4] Sifat linieritas pada fungsi ranking memenuhi sifat sebagai berikut:

1. $\Re(k\tilde{a}) = \mathrm{k}\Re(\tilde{a})$ untuk setiap skalar $k \in \mathbb{R}$.

2. $\Re(\tilde{a} \oplus \tilde{b}) = \Re(\tilde{a}) \oplus \Re(\tilde{b})$ untuk setiap $\tilde{a}, \tilde{b} \in F(\mathbb{R})$.

dengan $\tilde{a} = (a^l, a^u, \alpha, \beta)$, $\tilde{b} = (b^l, b^u, \gamma, \theta)$, dan $a^l, a^u, \alpha, \beta, b^l, b^u, \gamma, \theta \in \mathbb{R}$.

2.3. Program Linier *Fuzzy* Tidak Penuh

Program linier *fuzzy* dikatakan program linier *fuzzy* tidak penuh (PLFTP), dikarenakan terdapat variabel keputusan, koefisien fungsi tujuan, koefisien kendala atau ruas kanan kendala adalah bilangan-bilangan *fuzzy*. Adapun beberapa bentuk program linier *fuzzy* tidak penuh **[1]:**

1. Koefisien fungsi tujuan merupakan bilangan *fuzzy*

Memaksimalkan (atau meminimalkan) $Z = \sum_{j=1}^{n} \tilde{C}_j . x_j$

dengan kendala: $\sum_1^n a_{ij} x_j (\leq, =, \geq) b_i$, $i = 1,2, \dots, m$, $x_j \geq 0$ dimana $\tilde{C}_j$ merupakan koefisien fungsi tujuan bilangan *fuzzy*, $b_i \in \mathbb{R}^m$, $a_{ij} \in \mathbb{R}^{m\,x\,n}$.

2. Koefisien fungsi kendala merupakan bilangan *fuzzy*

Memaksimalkan (atau meminimalkan) $Z = \sum_{j=1}^{n} C_j x_j$

dengan kendala: $\sum_1^n \tilde{a}_{ij} x_j (\leq, =, \geq) b_i$, $i = 1,2, \dots, m$, $x_j \geq 0$ dimana $\tilde{a}_{ij}$ merupakan koefisien fungsi kendala bilangan *fuzzy*, $b_i \in \mathbb{R}^m$, $C_j \in \mathbb{R}^n$.

3. Ruas kanan fungsi kendala merupakan bilangan *fuzzy*

Memaksimalkan (atau meminimalkan) $Z = \sum_{j=1}^{n} C_j x_j$

dengan kendala: $\sum_1^n a_{ij} x_j (\leq, =, \geq) \tilde{b}_i$, $i = 1,2, \dots, m$, $x_j \geq 0$ dimana $\tilde{b}_i$ merupakan koefisien fungsi kendala bilangan *fuzzy*, $C_j \in \mathbb{R}^n$, $a_{ij} \in \mathbb{R}^{m\,x\,n}$.

2.4 Metode Simpleks dan Algoritma Determinan Matriks Ordo Dua[5]

1. Langkah 1 : menuliskan masalah dalam bentuk standar yaitu mengubah kendala ke dalam kesamaan-kesamaan. Jika kendala berbentuk ≤ maka persamaan kendala diubah dengan menambahkan variabel kelonggaran /*slack variable* (Si) pada ruas kiri. Jika kendala berbentuk ≥ maka persamaan kendala diubah dengan mengurangkan variabel surplus (Si) dan menambahkan variabel semu /*artificial variable* (Ri) pada ruas kiri. Jika kendala berbentuk = maka persamaan kendala diubah dengan

menambahkan variabel semu/ *artificial variable* (Ri) pada ruas kiri.

2. **Langkah 2 : menemukan solusi basis awal**. Ketika kendala berbentuk $\leq$, misalkan semua variabel non basis $(x_1, x_2, \dots, x_n)$ sama dengan nol $(x_j = 0)$ sehingga didapatkan $Si = bi$. Dan Si merupakan variabel basis awal. Ketika kendala berbentuk $\geq$, misalkan semua variabel non basis dan *surplus variable* sama dengan nol $(x_j = 0, Si = 0)$ sehingga didapatkan $Ri = bi$. Dan Ri merupakan variabel basis awal. Ketika kendala berbentuk =, misalkan semua variabel non basis sehingga didapatkan $Ri = bi$. Dan Ri merupakan variabel basis awal.
3. **Langkah 3 : Optimalitas tes**. Hitung $Z_j = \sum C_B a_{ij}$ dan $Z_j - C_j$. Jika $Z_j - C_j \geq 0$ untuk kasus maksimum dan jika $Z_j - C_j \leq 0$ untuk kasus minimum maka solusi optimal telah dicapai . Untuk sekurang-kurangnya satu j, dengan $Z_j - C_j < 0$ untuk kasus maksimum dan $Z_j - C_j > 0$ untuk kasus minimum, lanjutkan ke langkah 4.
4. **Langkah 4 : iterasi menuju solusi optimal menggunakan algoritma determinan matriks ordo dua.**

a. Memilih Entering Variabel (EV)

   Untuk Fungsi tujuan memaksimumkan, pilih kolom dengan $Z_j - C_j$ yang paling negatif dan untuk fungsi tujuan meminimumkan pilih kolom $Z_j - C_j$ yang paling positif (positif terbesar).

b. Memilih Leaving Variabel (LV)

   Hitung $\min\left\{\frac{x_B}{a_{ij}} \middle| a_{ij} > 0, a_{ij} \in kolom\ pivot\right\}$. Jika semua $a_{ij} < 0$, maka solusinya adalah unbounded.

c. Memperbaharui solusi baru/ penyusunan tabel simpleks menggunakan determinan.

   Misalkan elemen pivot adalah a_{ij}. Bagi baris pivot dengan elemen pivot. Semua elemen lain dalam kolom pivot harus dibuat nol.

   Kasus 1 : Untuk a_{rs} suatu elemen dalam matriks koefisien dengan $r < i,\ s < j$, elemen a_{rs} diperbaharui menjadi $det\begin{pmatrix} a_{rs} & a_{rj} \\ a_{is} & a_{ij} \end{pmatrix}$. Selanjutnya elemen a_{rs} distandarkan, yaitu membaginya dengan elemen pivot.

   Kasus 2 : Untuk a_{pq} suatu elemen dalam matriks koefisien dengan $p > i, q < j$, elemen a_{pq} diperbaharui menjadi $(-1).det\begin{pmatrix} a_{iq} & a_{ij} \\ a_{pq} & a_{pj} \end{pmatrix}$. Selanjutnya elemen a_{pq} distandarkan, yaitu membaginya dengan elemen pivot.

   Kasus 3 : Untuk a_{uv} suatu elemen dalam matriks koefisien dengan $u < i, v > j$,

elemen a_{uv} diperbaharui menjadi $(-1).det\begin{pmatrix} a_{uj} & a_{uv} \\ a_{ij} & a_{iv} \end{pmatrix}$. Selanjutnya elemen a_{uv} distandarkan, yaitu membaginya dengan elemen pivot.

Kasus 4 : Untuk a_{gh} suatu elemen dalam matriks koefisien dengan $g > i, h > j$, elemen a_{gh} diperbaharui menjadi $det\begin{pmatrix} a_{ij} & a_{ih} \\ a_{gj} & a_{gh} \end{pmatrix}$. Selanjutnya elemen a_{gh} distandarkan, yaitu membaginya dengan elemen pivot.

5. **Langkah 5** : Kembali ke langkah 3.

Prosedur optimasi masalah program linier fuzzy tidak penuh dengan koefisien fungsi tujuan merupakan bilangan trapezoidal *fuzzy* yaitu :

1. Memformulasikan masalah Program Linier *Fuzzy*.
2. Mengkonversikan masalah Program Linier *Fuzzy* ke masalah Program Linier Crisp menggunakan mentode penegasan *Robust Ranking*.
3. Optimasi masalah Program Linier Crisp menggunakan metode simpleks dan algoritma determinan matriks ordo dua.

Contoh. "*Family Group*" adalah *home industry* yang bergerak di bidang makanan yang terletak di Jalan Raya Km 12 Mungka, Payakumbuh. Family *Group* memproduksi berbagai jenis makanan diantaranya keripik bawang, keripik singkong balado dan keripik pisang. Untuk memproduksi keripik bawang, keripik singkong balado dan keripik pisang dibutuhkan 15 bahan baku, yaitu tepung terigu, tepung maizena, tepung perancis, tepung beras,telur, bawang putih, bawang merah, blueband, ajinomoto, cabe, garam, singkong, pisang mentah, gula dan vanille.

Komposisi bahan baku dalam dalam pembuatan 1 kg keripik bawang membutuhkan 0.833 kg tepung terigu, 0.016 kg tepung maizena, 0.083 kg tepung perancis, 0.041 kg tepung beras, 0.05 kg telur, 0.05 kg bawang putih, 0.05 kg bawang merah, 0.033 blueband, 0.0083 ajinomoto. Untuk membuat 1 kg keripik singkong balado membutuhkan 2.4 kg singkong, 0.12 kg cabe, 0.016 kg bawang putih, 0.002 kg garam, 0.008 kg ajinomoto. Untuk membuat 1 kg keripik pisang membutuhkan 5 kg pisang mentah, 0.125 kg gula, 0.01 kg garam, dan 0.0005 kg vanille.

Persediaan bahan baku setiap minggunya yaitu 40 kg tepung terigu, 1 kg tepung maizena, 5 kg tepung perancis, 2 kg tepung beras, 4 kg telur, 7 kg bawang putih 2.5 kg bawang merah,1.5 kg blueband, 3 kg ajinomoto, 30 kg cabe, 1 kg garam, 600 kg singkong, 220 kg pisang mentah, 6 kg gula, dan 0,1 kg vanile.

Keuntungan keripik bawang berkisar antara Rp 8000 – 10000. Keuntungan ini bisa mengalami kenaikan jika penjualan dilakukan keluar kota, tetapi tidak pernah mencapai Rp 20000 dan jika harga bahan baku naik keuntungan bisa mengalami penurunan tetapi tidak pernah mencapai Rp 4000. Keuntungan keripik singkong balado berkisar antara Rp 9000 – 15000. Keuntungan ini bisa mengalami kenaikan jika penjualan dilakukan keluar kota, tetapi tidak pernah mencapai Rp 25000 dan jika harga bahan baku naik keuntungan bisa mengalami penurunan tetapi tidak pernah mencapai Rp 7000. Keuntungan keripik pisang berkisar antara Rp 6000 – 8000. Keuntungan ini bisa mengalami kenaikan jika penjualan dilakukan keluar kota, tetapi tidak pernah mencapai Rp 18000 dan jika harga bahan baku naik keuntungan bisa mengalami penurunan tetapi tidak pernah mencapai Rp 3000.

Berdasarkan kondisi tersebut, berapakah keuntungan maksimum yang bisa didapat oleh *home industry* "*Family Group*"?

Penyelesaian :

1. Memformulasikan masalah Program Linier *Fuzzy*.

Variabel Keputusan :

x_1 = Jumlah keripik bawang yang diproduksi (dalam kg)

x_2 = Jumlah keripik singkong balado yang diproduksi (dalam kg)

x_3 = Jumlah keripik pisang yang diproduksi (dalam kg).

Kasus tersebut dapat diformulasikan sebagai berikut:

Maksimalkan $Z = (8000,10000,4000,10000)x_1 + (6000,15000,2000,10000)x_2 + (6000,8000,3000,10000)x_3$,

dengan kendala : $0.833x_1 \leq 40$; $0.016x_1 \leq 1$; $0.083x_1 \leq 5$; $0.041x_1 \leq 2$; $0.05x_1 \leq 4$; $0.05x_1 + 0.016x_2 \leq 7$; $0.05x_1 \leq 2.5$; $0.033x_1 \leq 1.5$; $0.0083x_1 + 0.008x_2 \leq 3$; $0.12x_2 \leq 30$; $0.002x_2 + 0.001x_3 \leq 1$; $2.4x_2 \leq 600$; $5x_3 \leq 220$; $0.125x_3 \leq 6$; $0.005x_3 \leq 0.1$; $x_1, x_2, x_3 \geq 0$.

2. Mengkonversikan masalah Program Linier *Fuzzy* ke masalah Program Linier Crisp menggunakan mentode penegasan Robust Ranking.

Untuk $\tilde{c}_1 = (8000,10000,4000,10000)$, $\Re(\tilde{c}_1) = 10500$.

Untuk $\tilde{c}_2 = (9000,15000,2000,10000)$, $\Re(\tilde{c}_2) = 14000$.

Untuk $\tilde{c}_3 = (6000,8000,3000,10000)$, $\Re(\tilde{c}_3) = 8750$.

Sehingga didapatkan Program Linier Crispnya yaitu :

Maksimalkan $Z = 10500x_1 + 14000x_2 + 8750x_3$

dengan kendala : $0.833x_1 \leq 40;\ 0.016x_1 \leq 1;\ 0.083x_1 \leq 5;\ 0.041x_1 \leq 2;\ 0.05x_1 \leq 4;$ $0.05x_1 + 0.016x_2 \leq 7;\quad 0.05x_1 \leq 2.5;\quad 0.033x_1 \leq 1.5;\qquad 0.0083x_1 + 0.008x_2 \leq 3;$ $0.12x_2 \leq 30;\quad 0.002x_2 + 0.001x_3 \leq 1;\quad 2.4x_2 \leq 600;\quad 5x_3 \leq 220;\quad 0.125x_3 \leq 6;$ $0.005x_3 \leq 0.1;\ x_1, x_{2,} x_3 \geq 0.$

3. Optimasi masalah Program Linier Crisp menggunakan metode simpleks dan algoritma determinan matriks ordo dua.

Langkah 1 : Tuliskan masalah dalam bentuk baku atau standar.

Maks $Z = 10500x_1 + 14000x_2 + 8750x_3 + 0s_1 + 0s_2 + 0s_3 + 0s_4 + 0s_5 + 0s_6 + 0s_7 + 0s_8 + 0s_9 + 0s_0 + 0s_{11} + 0s_{12} + 0s_{13} + 0s_{14} + 0s_{15}$

dengan kendala :

$0.833x_1 + s_1 = 40;\qquad 0.016x_1 + s_2 = 1;\ 0.083x_1 + s_3 = 5;\qquad 0.041x_1 + s_4 = 2;$ $0.05x_1 + s_5 = 4;\ 0.05x_1 + 0.016x_2 + s_6 = 7;\ 0.05x_1 + s_7 = 2.5;\ 0.033x_1 + s_8 = 1.5;$ $0.0083x_1 + 0.008x_2 + s_9 = 3;\quad 0.12x_2 + s_{10} = 30;\quad 0.002x_2 + 0.001x_3 + s_{11} = 1;$ $2.4x_2 + s_{12} = 600;\quad 5x_3 + s_{13} = 220;\quad 0.125x_3 + s_{14} = 6;\quad 0.005x_3 + s_{15} = 0.1;$ $x_1, x_{2,} x_3 \geq 0.$

Langkah 2 : Menentukan solusi basis awal .

$s_1 = 40, s_2 = 1, s_3 = 5, s_4 = 2, s_5 = 4, s_6 = 7, s_7 = 2{,}5\ ,\ s_8 = 1{,}5\ , s_9 = 3, s_{10} = 30,$ $s_{11} = 1, s_{12} = 600,\quad s_{13} = 220, s_{14} = 6\ dan\ s_{15} = 0{,}1.\quad s_i, i = 1{,}2, \ldots, 15$ merupakan variabel basis awal.

Langkah 3 : Optimalitas Tes.

Hitung $Z_j = \sum C_B a_{ij}$, sehingga tabel simpleks awal adalah sebagai berikut :

Tabel Simpleks Iterasi Awal

		C_j	10500	14000	8750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_B	x_B	y_B	x_1	x_2	x_3	s_1	s_2	s	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
0	s_1	40	0.833	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	s_2	1	0.016	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	s_3	5	0.083	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	s_4	2	0.041	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	s_5	4	0.05	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	s_6	7	0.05	0.016	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	s_7	2.5	0.05	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	s_8	1.5	0.033	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	s_9	3	0.0083	0.008	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	s_{10}	30	0	0.12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	s_{11}	1	0	0.002	0.01	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	s_{12}	600	0	2.4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	s_{13}	220	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	s_{14}	6	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	s_{15}	0.1	0	0	0.0005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		Z_j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		$Z_j - C_j$	-10500	-14000	-8750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Langkah 4 : Iterasi menuju solusi optimal menggunakan determinan

a. Memilih Entering Variabel

Pilih kolom dengan $Z_j - C_j = -14000$ menjadi kolom pivot (EV).

b. Memilih Leaving Variabel

Min $\left\{\frac{7}{0.016}; \frac{3}{0.008}; \frac{30}{0.12}; \frac{1}{0.002}; \frac{600}{2.4}\right\}$ = min $\{437.5; 375; 250; 500; 250\}$ = 250.

Sehingga didapatkan LV yaitu baris kesebelas. Dengan elemen pivot adalah 0.12. Dan x_2 menjadi variabel basis.

c. Memperbaharui Solusi Baru / Penyusunan Tabel Simpleks Menggunakan Determinan

Baris pivot dibagi dengan elemen pivot dan elemen lain dalam kolom pivot dibuat menjadi nol (0). Elemen-elemen yang lain diperbaharui menggunakan determinan matriks berdasarkan empat kasus sebelumnya. Sehingga didapatkan :

Tabel Simpleks Iterasi 1

		C_j	10500	14000	8750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_B	x_B	y_B	x_1	x_2	x_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
0	s_1	40	0.833	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	s_2	1	0.016	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	s_3	5	0.083	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	s_4	2	0.041	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	s_5	4	0.05	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	s_6	3	0.05	0	0	0	0	0	0	0	1	0	0	0	$-\frac{0,004}{0,03}$	0	0	0	0	0
0	s_7	2.5	0.05	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	s_8	1.5	0.033	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	s_9	1	0.0083	0	0	0	0	0	0	0	0	0	0	1	$-\frac{0,002}{0,03}$	0	0	0	0	0
14000	x_2	250	0	1	0	0	0	0	0	0	0	0	0	0	$\frac{0,25}{0,03}$	0	0	0	0	0
0	s_{11}	0.5	0	0	0.01	0	0	0	0	0	0	0	0	0	$-\frac{0,001}{0,006}$	1	0	0	0	0
0	s_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	-20	0	1	0	0	0
0	s_{13}	220	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	s_{14}	6	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	s_{15}	0.1	0	0	0.0005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		Z_j	0	14000	0	0	0	0	0	0	0	0	0	0	$\frac{3500}{0,003}$	0	0	0	0	0
		$Z_j - C_j$	-10500	0	-8750	0	0	0	0	0	0	0	0	0	$\frac{3500}{0,003}$	0	0	0	0	0

Karena terdapat $Z_j - C_j < 0$, maka solusi belum optimal. Sehingga perlu dilakukan iterasi lagi sesuai dengan langkah sebelumnya.

Pada iterasi selanjutnya didapatkan tabel simpleks yaitu sebagai berikut :

Tabel Simpleks Iterasi 2

		C_j	10500	14000	8750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_B	x_B	y_B	x_1	x_2	x_3	s_1	s_2	s	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
0	s_1	$\frac{0,0705}{0,033}$	0	0	0	1	0	0	0	0	0	0	$-\frac{0,833}{0,033}$	0	0	0	0	0	0	0
0	s_2	$\frac{0,009}{0,033}$	0	0	0	0	1	0	0	0	0	0	$-\frac{0,016}{0,033}$	0	0	0	0	0	0	0
0	s_3	$\frac{0,0405}{0,033}$	0	0	0	0	0	1	0	0	0	0	$-\frac{0,083}{0,033}$	0	0	0	0	0	0	0
0	s_4	$\frac{0,0045}{0,033}$	0	0	0	0	0	0	1	0	0	0	$-\frac{0,041}{0,033}$	0	0	0	0	0	0	0
0	s_5	$\frac{0,057}{0,033}$	0	0	0	0	0	0	0	1	0	0	$-\frac{0,05}{0,033}$	0	0	0	0	0	0	0
0	s_6	$\frac{0,024}{0,033}$	0	0	0	0	0	0	0	0	1	0	$-\frac{0,05}{0,033}$	0	$-\frac{0,004}{0,03}$	0	0	0	0	0
0	s_7	$\frac{0,0075}{0,033}$	0	0	0	0	0	0	0	0	0	1	$-\frac{0,05}{0,033}$	0	0	0	0	0	0	0
10500	x_1	$\frac{0,5}{0,011}$	1	0	0	0	0	0	0	0	0	0	$\frac{1}{0,033}$	0	0	0	0	0	0	0
0	s_9	$\frac{0,02055}{0,033}$	0	0	0	0	0	0	0	0	0	0	$-\frac{0,0083}{0,033}$	1	$-\frac{0,002}{0,03}$	0	0	0	0	0
14000	x_2	250	0	1	0	0	0	0	0	0	0	0	0	0	$\frac{0,25}{0,03}$	0	0	0	0	0
0	s_{11}	0.5	0	0	0.01	0	0	0	0	0	0	0	0	0	$-\frac{0,001}{0,006}$	1	0	0	0	0
0	s_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	-20	0	1	0	0	0
0	s_{13}	220	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	s_{14}	6	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	s_{15}	0.1	0	0	0.0005	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		Z_j	10500	14000	0	0	0	0	0	0	0	0	$\frac{10500}{0,033}$	0	$\frac{3500}{0,03}$	0	0	0	0	0
		$Z_j - C_j$	0	0	-8750	0	0	0	0	0	0	0	$\frac{10500}{0,033}$	0	$\frac{3500}{0,03}$	0	0	0	0	0

Tabel Simpleks Iterasi 3

		C_j	10500	14000	8750	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_B	x_B	y_B	x_1	x_2	x_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
0	s_1	$\frac{0,0705}{0,033}$	0	0	0	1	0	0	0	0	0	0	$-\frac{0,833}{0,033}$	0	0	0	0	0	0	0
0	s_2	$\frac{0,009}{0,033}$	0	0	0	0	1	0	0	0	0	0	$-\frac{0,016}{0,033}$	0	0	0	0	0	0	0
0	s_3	$\frac{0,0405}{0,033}$	0	0	0	0	0	1	0	0	0	0	$-\frac{0,083}{0,033}$	0	0	0	0	0	0	0
0	s_4	$\frac{0,0045}{0,033}$	0	0	0	0	0	0	1	0	0	0	$-\frac{0,041}{0,033}$	0	0	0	0	0	0	0
0	s_5	$\frac{0,057}{0,033}$	0	0	0	0	0	0	0	1	0	0	$-\frac{0,05}{0,033}$	0	0	0	0	0	0	0
0	s_6	$\frac{0,024}{0,033}$	0	0	0	0	0	0	0	0	1	0	$-\frac{0,05}{0,033}$	0	$-\frac{0,004}{0,03}$	0	0	0	0	0
0	s_7	$\frac{0,0075}{0,033}$	0	0	0	0	0	0	0	0	0	1	$-\frac{0,05}{0,033}$	0	0	0	0	0	0	0
10500	x_1	$\frac{0,5}{0,011}$	1	0	0	0	0	0	0	0	0	0	$\frac{1}{0,033}$	0	0	0	0	0	0	0
0	s_9	$\frac{0,02055}{0,033}$	0	0	0	0	0	0	0	0	0	0	$-\frac{0,0083}{0,033}$	1	$-\frac{0,002}{0,03}$	0	0	0	0	0
14000	x_2	250	0	1	0	0	0	0	0	0	0	0	0	0	$\frac{0,25}{0,03}$	0	0	0	0	0
0	s_{11}	0,06	0	0	0	0	0	0	0	0	0	0	0	0	$-\frac{0,001}{0,006}$	1	0	-0,002	0	0
0	s_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	-20	0	1	0	0	0
8750	x_3	44	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0,2	0	0
0	s_{14}	0,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0,025	1	0
0	s_{15}	0,078	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0,0001	0	1
		Z_j	10500	14000	8750	0	0	0	0	0	0	0	$\frac{10500}{0,033}$	0	$\frac{3500}{0,03}$	0	0	1750	0	0
		$Z_j - C_j$	0	0	0	0	0	0	0	0	0	0	$\frac{10500}{0,033}$	0	$\frac{3500}{0,03}$	0	0	1750	0	0

Berdasarkan tabel diatas tersebut terlihat bahwa semua $Z_j - C_j \geq 0$, maka iterasi dihentikan. Sehingga diperoleh solusi optimalnya yaitu

$(x_1; x_2; x_3) = \left(\frac{0.5}{0.011}; 250; 44\right)$ = (45.45455;250;44) dengan :

$Z = (10500 \times 45.45455) + (14000 \times 250) + (8750 \times 44)$

$= 477273 + 3500000 + 385000$

$= 4362273.$

Jadi keuntungan maksimum yang didapatkan oleh *home industry Family Group* adalah sebesar $Rp\ 4{,}362{,}273.00$ dengan jumlah keripik bawang yang harus diproduksi sebanyak 45.45455 kg, keripik singkong balado sebanyak 250 kg dan keripik pisang sebanyak 44 kg.

III. KESIMPULAN

Metode Simpleks yang melibatkan algoritma determinan matriks ordo dua dengan metode simpleks yang melibatkan operasi baris elementer memiliki perbedaan yang terletak pada iterasi, yaitu dalam memperbaharui solusi baru atau penyusunan tabel simpleks. Solusi optimal yang didapatkan dari kedua metode adalah sama. Metode simpleks yang melibatkan algoritma determinan matriks ordo dua ini lebih efisien karena pada setiap langkah iterasi untuk memperbaharui solusi baru bisa langsung ditentukan perubahan terhadap setiap elemen-elemen dalam tabel simpleks dengan menghitung determinan matriks ordo dua. Sedangkan dengan menggunakan operasi baris elementer harus ditentukan dulu operasi untuk setiap baris, setelah itu baru dilakukan perubahan-perubahan untuk elemen-elemen dalam tabel simpleks.

IV. DAFTAR PUSTAKA

[1] Iden H.A., Farrah Alaa A., 2014. *Ranking Function Methods For Solving Fuzzy Linear Programming Problems,* Mathematical Theory and Modeling, Vol.4, No.4, pp:65-72.

[2] K. Usha Madhuri, B. PardhaSaradhi, dan N. Ravi Shankar, 2013. *Fuzzy Linear Programming Model for Critical Path Analysis.* International Journal Contemporer Mathematic Science, Vol. 8, pp: 93 – 116.

[3] P. Jayaraman, R Jahirnussian. 2013. *Fuzzy Optimal Transportation Problems by Improved Zero Suffix Method via Robust Rank Techniques*. International Journal of Fuzzy Mathematics and Systems, Vol 3, No 4, pp : 301-311.

[4] S.H Nasseri, E Ardil. 2009. *Simplex Method for Fuzzy Variable Linear Programming Problems*. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, Vol 3, No 10.

[5] Jervin Zen Lobo. 2015. *Two Square Determinant Approach for Simplex Method*. IOSR Journal of Mathematics, Vol 11, Issue 5, Ver IV, pp : 01-04.