

# Penerapan Fuzzy Integer Programming untuk Optimasi Kandungan Gizi Menu Harian

Galang Gilang Ramadhan<sup>a</sup>, Lailil Muflikhah<sup>b</sup>, Marji<sup>c</sup>

<sup>a</sup>Mahasiswa, <sup>b</sup>Dosen Pembimbing 1, <sup>c</sup>Dosen Pembimbing 2  
Program Studi Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang  
Email: [galanggilangramadhan@gmail.com](mailto:galanggilangramadhan@gmail.com)

## Abstract

Daily menu nutrition optimization's main goal is to help user find the cheapest variation of their daily menu list, according to their dietary needs and also specified cost constraint. This problem is quite challenging due to many imprecise data are involved so it can't be solved by using basic optimization method like Linear Programming. Fuzzy Linear Programming is another promising solution. Though, due to its linear nature, this method has an effect that can resulting an unpalatable food combination. Fuzzy Integer Programming (FIP), which extends Integer Programming method, apply stricter constraints to the problem's mathematical model. By using a method that converts an FIP model into Multiple Objective Integer Programming Problem (MOIPP) model, Integer Programming with fuzzy data is easily solvable. A test that try to comparing the result of this method with an a priori optimal result (obtained from an exhaustive combination generation) shows that this method is capable to find the optimal menu variation, within various data fuzziness settings, with optimality of 100%. Hence, Fuzzy Integer Programming can be considered as a reliable solution to solve daily menu nutrition optimization problem.

Keywords: Fuzzy Integer Programming; Daily Menu; Optimization; Diet Problem;

## Abstrak

Tujuan utama optimasi kandungan gizi menu harian adalah untuk membantu pengguna menemukan variasi termurah dari daftar menu harian mereka, berdasarkan kebutuhan gizinya serta batasan biaya yang ditetapkan. Permasalahan ini cukup menantang karena banyak melibatkan data yang sifatnya tak pasti atau buram sehingga tidak dapat diselesaikan dengan metode optimasi dasar seperti *Linear Programming*. *Fuzzy Linear Programming* adalah solusi lain yang dapat digunakan. Walaupun demikian, dikarenakan sifatnya yang linear, metode ini menimbulkan efek di mana kombinasi makanan yang dihasilkan bisa jadi tidak enak untuk dikonsumsi (*unpalatable*). *Fuzzy Integer Programming (FIP)*, yang merupakan pengembangan dari *Integer Programming*, menerapkan batasan-batasan yang lebih ketat pada model matematika dari permasalahan. Dengan menggunakan metode yang mengonversi model FIP menjadi model *Multiple Objective Integer Programming Problem (MOIPP)*, *Integer Programming* yang melibatkan data tak pasti dapat dipecahkan dengan mudah. Pengujian dengan membandingkan hasil dari metode ini dengan sebuah hasil *a priori optimal* (didapatkan dari pembangkitan seluruh kombinasi yang mungkin) menunjukkan bahwa metode ini mampu menemukan variasi menu optimal dengan tingkat optimalitas 100% dalam berbagai kondisi keburaman data. Dengan demikian, *Fuzzy Integer Programming* dapat dikatakan sebagai solusi yang dapat diandalkan untuk menyelesaikan permasalahan optimasi kandungan gizi menu harian.

Kata kunci: Fuzzy Integer Programming; Menu Harian; Optimasi; Diet Problem;

## 1. Pendahuluan

Seiring dengan semakin meningkatnya harga bahan-bahan pangan, masyarakat dituntut untuk semakin pandai dalam menyiasati menu harian mereka. Namun karena batasan biaya, seringkali kandungan gizi pada menu harian dikesampingkan. Padahal, dengan melakukan variasi pada susunan menu harian yang ada, akan sangat mungkin untuk mendapatkan susunan menu dengan kandungan gizi yang cukup dan biaya yang lebih terjangkau. Masalahnya adalah, kebanyakan masyarakat awam tidak memiliki waktu dan sumber daya yang cukup untuk mempertimbangkan variasi-variasi tersebut.

Tujuan utama optimasi kandungan gizi menu harian adalah untuk membantu pengguna menemukan variasi termurah dari daftar menu harian mereka, berdasarkan

kebutuhan gizinya serta batasan biaya yang ditetapkan. Permasalahan ini cukup menantang karena banyak melibatkan data yang sifatnya tak pasti atau buram sehingga tidak dapat diselesaikan dengan metode optimasi dasar seperti *Linear Programming*. Untuk dapat mengakomodasi keburaman data tersebut, dapat dipergunakan metode *Fuzzy Linear Programming*. Namun dikarenakan sifatnya yang linear, metode ini juga memiliki kekurangan di mana kombinasi makanan yang dihasilkan bisa jadi tidak enak untuk dikonsumsi (*unpalatable*). Efek ini telah coba diminimalisasi pada penelitian yang dilakukan oleh Prasetyo (2010) dengan memanfaatkan Ukuran Rumah Tangga (URT), yakni ukuran normal konsumsi bahan makanan, sebagai batasan. Namun, solusi

ini dirasa masih kurang efektif mengingat URT tidak menggambarkan proporsi suatu bahan relatif terhadap bahan lainnya pada suatu menu.

*Fuzzy Integer Programming (FIP)* adalah metode lain yang dapat diterapkan pada kasus optimasi kandungan gizi menu harian. Metode yang merupakan pengembangan dari *Integer Programming* ini memiliki kelebihan di mana metode ini menerapkan batasan-batasan yang lebih ketat pada model matematika dari permasalahan yang akan diselesaikan. Metode ini juga dapat mengakomodasi data-data buram dengan cara mengonversi model permasalahan yang akan diselesaikan menjadi model *Multiple Objective Integer Programming Problem (MOIPP)* (Dash & Dash, 2012). FIP telah diterapkan oleh Oruç, et al. (2012) sebagai algoritma perencanaan menu bagi pekerja. Pada penelitian tersebut, FIP digunakan untuk merencanakan menu selama 20 hari dari kerangka menu yang telah ditentukan dengan memperhatikan batasan kebutuhan gizi pekerja. Namun demikian, model *non-selective menu planning* (pengguna tidak memilih sendiri menunya) seperti yang digunakan oleh Oruç, et al. (2012) memiliki kelemahan dari sisi fleksibilitas pengguna.

Pada penelitian ini, akan digunakan pendekatan yang hampir sama dengan yang digunakan oleh Oruç, et al. (2012). Akan tetapi, model perencanaan menu yang akan digunakan adalah *selective menu planning*. Artinya, sistem tidak memilihkan menu untuk pengguna, tetapi pengguna sendiri yang memilih menu-menu yang ingin dioptimasi. Perbedaan lain antara penelitian ini dengan penelitian oleh Oruç, et al. (2012) terletak pada obyek yang ingin dioptimasi. Berbeda dengan pendekatan Oruç, et al. (2012) yang pada dasarnya adalah optimasi di tingkat bahan, penelitian ini akan melakukan optimasi di tingkat menu. Sebuah menu dapat terdiri dari beberapa bahan, dan bahan-bahan dalam sebuah menu independen terhadap bahan-bahan dari menu lain. Setelah pengguna memilih menu-menu yang ingin dioptimasi, FIP akan melakukan

optimasi pada jumlah penyajian tiap menu. Dengan melakukan optimasi pada tingkat menu dan bukan pada tingkat bahan, komposisi bahan akan tetap proporsional terhadap resep aslinya. Proporsionalitas komposisi bahan inilah yang dapat menjaga makanan tetap *palatable*.

### 1.1. Menu harian

Menu harian adalah komposisi makanan yang dikonsumsi seseorang atau sebuah keluarga dalam satu hari. Idealnya, kandungan gizi dalam menu harian harus mencukupi kebutuhan gizi orang atau keluarga tersebut. Di Indonesia dikenal slogan "4 sehat 5 sempurna" yang dapat digunakan masyarakat awam sebagai panduan untuk menyusun menu harian yang seimbang dan berkecukupan. Komposisi nasi, lauk, sayur, buah dan susu dalam menu "4 sehat 5 sempurna" diperkirakan dapat memenuhi kebutuhan energi, protein, lemak, vitamin serta mineral seseorang secara umum. Pada Tabel 1 diberikan contoh menu harian untuk 5 hari.

Saat ini, slogan "4 sehat 5 sempurna" dianggap sudah tidak relevan lagi. Hal ini disebabkan oleh timbulnya salah kaprah di masyarakat yang justru berujung pada masalah gizi seperti obesitas dan kegemukan. Dengan semakin berkembangnya ilmu gizi, konsep lama ini akhirnya digantikan oleh konsep atau pola "gizi seimbang". Pola gizi seimbang menekankan pada pengaturan porsi atau jumlah nutrisi-nutrisi yang perlu dikonsumsi. Pengaturan ini disesuaikan dengan kebutuhan tiap-tiap individu, berdasarkan golongan usia, jenis kelamin, serta tingkat aktivitas individu tersebut. Sejalan dengan hal tersebut, maka penyusunan menu harian pun perlu dilakukan dengan perhitungan yang lebih cermat agar didapatkan komposisi menu harian dengan kandungan gizi yang sesuai dengan kebutuhan.

Hari Ke-	Makanan Pokok	Lauk	Sayur	Buah	Selangan
1	Nasi	Tempe Goreng	Lodeh	Pepaya	Singkong Goreng
2	Nasi	Pepes Ikan	Lalap	Pisang	Ubi Rebus
3	Nasi	Ikan Asin	Sayur Bayam	Jeruk	Jagung Rebus
4	Nasi	Tahu Goreng	Gado-Gado	Pisang	Singkong Rebus
5	Nasi	Telur Rebus	Tumis Kangkung	Pepaya	Kue Dari Singkong

Tabel 1 Contoh menu harian (Karyadi & Muhilal, 1985)

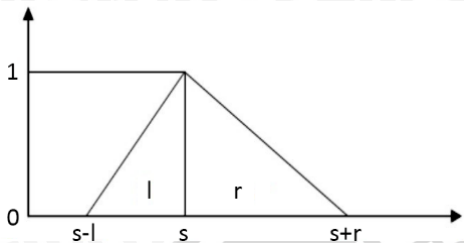
### 1.2. Fuzzy Number

Sebuah bilangan buram atau *fuzzy number* adalah kuantitas yang nilainya tidak tepat atau tidak pasti (Joshi, 2008). *Fuzzy number* tidak seperti bilangan biasa yang hanya merujuk pada satu nilai, melainkan merujuk pada

sekumpulan nilai-nilai yang mungkin. Tiap nilai yang mungkin memiliki bobot tersendiri yang disebut sebagai fungsi keanggotaan atau *membership function*, antara 0 dan 1. Salah satu tipe *fuzzy number* adalah *Triangular Fuzzy Number*, yang dinotasikan sebagai  $\tilde{A} = (s, l, r)$  dan memiliki fungsi keanggotaan (Fullér, 1991):

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 - \frac{s-x}{l}, & \text{untuk } s-l \leq x \leq s \\ 1 - \frac{x-s}{r}, & \text{untuk } s \leq x \leq s+r \\ 0, & \text{untuk yang lain} \end{cases}$$

Gambar 1 merupakan grafik fungsi keanggotaan dari sebuah *Triangular Fuzzy Number*.



Gambar 1 *Triangular Fuzzy Number* (Roseline & Amirtharaj, 2012)

Jika  $\tilde{A} = (s_1, l_1, r_1)$  dan  $\tilde{B} = (s_2, l_2, r_2)$  adalah dua buah *Triangular Fuzzy Number*, maka berlaku operasi-operasi berikut (Roseline & Amirtharaj, 2012):

- $\tilde{A} + \tilde{B} = (s_1, l_1, r_1) + (s_2, l_2, r_2)$   
 $= (s_1 + s_2, l_1 + l_2, r_1 + r_2)$
- $\tilde{A} - \tilde{B} = (s_1, l_1, r_1) - (s_2, l_2, r_2)$   
 $= (s_1 - s_2, l_1 - l_2, r_1 - r_2)$
- $x > 0, \tilde{A}x = (s_1x, l_1x, r_1x)$   
 $x < 0, \tilde{A}x = (r_1x, l_1x, s_1x)$
- $A \leq B$  jika dan hanya jika  $\text{MAX}(A, B) = B$

Sehingga untuk setiap *Triangular Fuzzy Number*  $\tilde{A} =$

$(s_1, l_1, r_1)$  dan  $\tilde{B} = (s_2, l_2, r_2)$ ,

$\tilde{A} \leq \tilde{B}$  jika dan hanya jika :

- 1)  $s_1 \leq s_2$
- 2)  $s_1 - l_1 \leq s_2 - l_2$
- 3)  $s_1 + r_1 \leq s_2 + l_2$

### 1.3. Integer Programming

*Integer Programming (IP)* adalah *Linear Programming* dengan tambahan batasan yang mengharuskan beberapa atau semua variabel ada dalam bentuk *integer* (Bosch & Trick, 2014). *Integer Programming* diselesaikan dengan terlebih dahulu melakukan relaksasi pada permasalahan. Relaksasi dari sebuah permasalahan *Integer Programming* adalah sebuah permasalahan baru sedemikian hingga :

- a) Setiap solusi bagi IP bersesuaian dengan sebuah solusi feasible bagi masalah hasil relaksasi.
- b) Setiap solusi bagi IP memiliki nilai fungsi objektif lebih besar atau sama dengan solusi yang bersesuaian pada masalah hasil relaksasi.

Setelah dilakukan relaksasi, permasalahan akan diselesaikan secara *branch-and-bound* sebagai berikut:

- **Langkah 0 (inisiasi)**

Tetapkan nilai fungsi tujuan *incumbent*  $v = \infty$  (untuk mengasumsikan kondisi awal di mana belum ada solusi *integer* yang memenuhi). Tetapkan penghitung *node* aktif  $k = 1$  dan tetapkan permasalahan awal sebagai *node* "aktif". Lanjutkan ke Langkah 1.

- **Langkah 1**

Jika  $k = 0$  (tidak ada *node* aktif), berhenti. Solusi *incumbent* adalah solusi optimal. Jika tidak ada solusi *incumbent*, misal  $v = \infty$ , maka permasalahan ini tidak memiliki solusi *integer*. Jika  $k \geq 1$  (masih ada *node* aktif), lanjutkan ke Langkah 2.

- **Langkah 2**

Pilih salah satu *node* aktif dan namai sebagai *current node*. Selesaikan permasalahan hasil relaksasi dari *current node*, dan tandai *current node* sebagai inaktif. Jika tidak ada solusi *feasible*, lanjutkan ke Langkah 3. Jika solusi dari *current node* memiliki nilai fungsi tujuan  $z^* \geq v$ , lanjutkan ke Langkah 4. Jika solusi semuanya *integer* dan  $z^* < v$ , lanjutkan ke Langkah 5. Selain itu, lanjutkan ke Langkah 6.

- **Langkah 3**

*Fathomed by Infeasibility*. Kurangi  $k$  dengan 1 dan kembali ke Langkah 1.

- **Langkah 4**

*Fathomed by Bound*. Kurangi  $k$  dengan 1 dan kembali ke Langkah 1.

- **Langkah 5**

*Fathomed by Integrality*. Ganti solusi *incumbent* dengan solusi *current node*. Tetapkan  $v = z^*$ , kurangi  $k$  dengan 1 dan kembali ke Langkah 1.

- **Langkah 6**

Buat cabang pada *current node*. Pilih variabel yang bernilai fraksional dari solusi permasalahan hasil relaksasi bagi *current node*. Namai variabel ini sebagai  $x_s$  dan namai nilai solusi optimalnya sebagai  $f$ . Buat dua *node* aktif baru pada *current*

*node*: satu dengan menambahkan batasan  $x_s \leq [f]$  dan yang lain dengan menambahkan batasan  $x_s \geq [f]$ . Tambahkan 1 pada  $k$  (2 *node* aktif baru minus 1 pencabangan pada *current node*) dan kembali ke Langkah 1.

#### 1.4. Konversi FIPP Ke MOIPP

Anggap terdapat FIPP (*Fuzzy Integer Programming Problem*) dengan koefisien fungsi tujuan dan koefisien batasan-batasannya berupa *fuzzy number*, atau dapat dituliskan sebagai :

$$\text{Max} \sum_{j=1}^n \tilde{C}_j x_j \quad (1)$$

Sedemikian hingga :

$$\sum_{x \geq 0} (S_{ij}, l_{ij}, r_{ij}) x_{ij} \leq (t_i, u_i, v_i) \quad (2a)$$

$$0 \leq i \leq m, 0 \leq j \leq n \quad (2b)$$

Permasalahan tersebut dapat dikonversi ke bentuk MOIPP (*Multiple Objective Integer Programming Problem*) sebagai (Dash & Dash, 2012):

$$\text{Max}_{x \in X} \sum_{m=1}^k f_m(x) \quad (3)$$

Sedemikian hingga :

$$\sum_{j=1}^n s_{ij} x_j \leq t_i \quad (4a)$$

$$\sum_{j=1}^n (s_{ij} - l_{ij}) x_j \leq t_i - u_i \quad (4b)$$

$$\sum_{j=1}^n (s_{ij} + r_{ij}) x_j \leq t_i + v_i \quad (4c)$$

## 2. Contoh Permasalahan

Misalkan daftar menu harian yang akan dioptimasi berisi menu-menu: “Soto Ayam”, “Nasi” dan “Jus Alpokat”. Data dari ketiga menu tersebut dapat dilihat pada Lampiran A.

### 2.1. Perhitungan nutrisi menu

Pada tahap ini, tiap menu akan dinormalisasi menjadi 1 porsi terlebih dahulu. Misalnya menu Soto Ayam adalah untuk 4 porsi, maka untuk tiap porsinya terdiri dari:

- Ayam :  $\frac{1}{4} \times (150, 10, 20) = (37.5, 2.5, 5)$

- Kol :  $\frac{1}{4} \times (200, 50, 50) = (50, 12.5, 12.5)$
- Bihun :  $\frac{1}{4} \times (50, 5, 5) = (12.5, 1.25, 1.25)$
- Telur :  $\frac{1}{4} \times (90, 50, 10) = (22.5, 12.5, 2.5)$
- Tomat :  $\frac{1}{4} \times (200, 5, 10) = (50, 1.25, 2.5)$

Selanjutnya, dilakukan perhitungan kandungan gizi untuk masing-masing bahan dari tiap menu dan menjumlahkannya menjadi kandungan gizi menu. Sebagai contoh, dari basis data didapatkan kandungan energi ayam per 100 gram adalah 302 *Kcal*, maka untuk mencari nilai energi (dalam *Kcal*) bahan ayam dari menu Soto Ayam perhitungannya adalah sebagai berikut:

$$\frac{1}{100} \times 302 \times (37.5, 2.5, 5) = (113.25, 7.55, 15.1)$$

Jika diasumsikan nutrisi yang diikutsertakan adalah energi, protein, lemak, vitamin A, vitamin B, dan vitamin C, maka dengan cara yang sama didapatkan data kandungan gizi tiap menu seperti pada Lampiran B.

### 2.2. Pembentukan persamaan FIPP

Tahap pertama pembentukan persamaan FIPP adalah pembentukan persamaan batasan. Misalkan terdapat Menu  $M1$  dan  $M2$  dengan porsi penyajian normal masing-masing  $(1, \frac{1}{2}, \frac{1}{4})$  dan  $(1, \frac{1}{4})$ , maka akan terbentuk 2 persamaan:

$$\begin{aligned} 1) \quad & a_{11}M11 + a_{12}M12 + a_{13}M13 + a_{14}M21 + \\ & a_{15}M22 \geq b_1 \\ 2) \quad & a_{21}M11 + a_{22}M12 + a_{23}M13 + a_{24}M21 + \\ & a_{25}M22 \geq b_2 \end{aligned}$$

Di mana  $M11 = 1$  porsi  $M1$ ,  $M12 = \frac{1}{2}$  porsi  $M1$ ,  $M13 = \frac{1}{4}$  porsi  $M1$ ,  $M21 = 1$  porsi  $M2$  dan  $M22 = \frac{1}{4}$  porsi  $M2$ .  $b_1$  dan  $b_2$  adalah angka batasan nutrisi atau dapat dikatakan kebutuhan gizi subjek. Diasumsikan kebutuhan gizi subjek adalah sebagai berikut:

- Energi = 1000 *Kcal*
- Protein = 25 *g/d*
- Vitamin A = 450  $\mu\text{g/d}$
- Vitamin B = 0.5 *mg/d*
- Vitamin C = 45 *mg/d*

Untuk membuat batasan menjadi lebih fleksibel, batasan akan diubah menjadi sebuah *triangular fuzzy number*, dengan nilai  $l$  dan  $r$  diatur oleh parameter yang disebut sebagai ‘*reference fuzziness*’. Parameter ini menentukan berapa persen nilai *left spread* ( $l$ ) dan/atau *right spread* ( $r$ ) dari nilai pastinya ( $s$ ). Pada contoh ini,

parameter *reference tolerance* akan diberi nilai 15%. Maka, untuk batasan energi dengan angka 1000 akan menghasilkan:

- *Left spread* :  $15\% \times 1000 = 150$
- *Right spread* :  $15\% \times 1000 = 150$
- *Triangular Fuzzy number* ( $s, l, r$ ) :  
(1000, 150, 150)

Dengan cara yang sama, batasan-batasan untuk nutrisi lain dapat dibentuk dan akan menghasilkan persamaan-persamaan seperti pada Lampiran C. Selain batasan-batasan tersebut, diperlukan lagi beberapa batasan tambahan. Batasan tambahan pertama adalah batasan untuk menjaga total harga agar tidak melebihi anggaran:

$$\begin{aligned} &(16000, 1000, 1000)M_{11} + \\ &(8000, 500, 500)M_{12} + (4000, 250, 250)M_{13} + \\ &(2000, 125, 125)M_{14} + (4000, 500, 500)M_{21} + \\ &(6000, 500, 500)M_{31} + (3000, 250, 250)M_{32} \leq \\ &(30000, 0, 0) \end{aligned}$$

Beberapa batasan juga diperlukan untuk memastikan setiap menu dalam menu set yang dipilih disajikan tepat 1 kali:

- 1)  $M_{11} + M_{12} + M_{13} + M_{14} = 1$
- 2)  $M_{21} = 1$
- 3)  $M_{31} + M_{32} = 1$

Batasan terakhir yang dibutuhkan adalah apa yang disebut sebagai '*integrality constraints*'. Batasan ini menjaga agar tiap variabel tujuan bernilai lebih dari atau sama dengan nol dan merupakan sebuah bilangan *integer*:

$$\begin{aligned} &M_{11}, M_{12}, M_{13}, M_{14}, M_{21}, M_{31}, M_{32} \geq 0, \\ &\text{integer} \end{aligned}$$

Setelah semua persamaan batasan tersebut terbentuk, tahap terakhir pembentukan persamaan FIPP adalah membentuk persamaan fungsi tujuan.

$$\begin{aligned} \text{Min. } Z &= 2 \times (8000, 500, 500)M_{11} + \\ &1 \times (8000, 500, 500)M_{12} + \\ &\frac{1}{2} \times (8000, 500, 500)M_{13} + \\ &\frac{1}{4} \times (8000, 500, 500)M_{14} + \\ &1 \times (4000, 500, 500)M_{21} + \\ &1 \times (6000, 500, 500)M_{31} + \\ &\frac{1}{2} \times (6000, 500, 500)M_{32} \end{aligned}$$

Atau

$$\begin{aligned} \text{Min. } Z &= (16000, 1000, 1000)M_{11} + \\ &(8000, 500, 500)M_{12} + (4000, 250, 250)M_{13} + \\ &(2000, 125, 125)M_{14} + (4000, 500, 500)M_{21} + \\ &(6000, 500, 500)M_{31} + (3000, 250, 250)M_{32} \quad (5) \end{aligned}$$

### 2.3. Konversi FIPP ke MOIPP

FIPP yang telah terbentuk selanjutnya akan dikonversi ke bentuk MOIPP agar dapat diselesaikan dengan menggunakan metode *Integer Programming* biasa. Berdasarkan Persamaan (3), fungsi tujuan pada Persamaan (5) dapat dikonversi menjadi :

$$\begin{aligned} \text{Min. } Z &= 18000M_{11} + 9000M_{12} + 4500M_{13} + \\ &2250M_{14} + 5000M_{21} + 7000M_{31} + \\ &3500M_{32} \quad (6) \end{aligned}$$

Konversi juga dilakukan pada batasan-batasan FIPP. Berdasarkan Persamaan (4a) hingga Persamaan (4c), sebuah persamaan batasan yang melibatkan *fuzzy number* dalam FIPP akan dikonversi menjadi 3 buah persamaan batasan. Hasil dari konversi ini adalah persamaan-persamaan dalam Lampiran D.

### 2.4. Pemecahan dengan integer programming

Setelah fungsi tujuan dan batasan-batasan dikonversi, maka permasalahan tersebut dapat dipecahkan dengan metode *Integer Programming* biasa. Permasalahan *Integer Programming* dengan fungsi tujuan Persamaan (6) dan batasan-batasan pada Lampiran D menghasilkan solusi:

$$\begin{aligned} &(M_{11}, M_{12}, M_{13}, M_{14}, M_{21}, M_{31}, M_{32}) = \\ &(0, 0, 0, 1, 1, 1, 0) \end{aligned}$$

Solusi dari *Integer Programming* ini juga sekaligus merupakan solusi akhir dari *Fuzzy Integer Programming*. Dari solusi tersebut diketahui bahwa menu-menu yang dipilih adalah  $M_{14}$ ,  $M_{21}$  dan  $M_{31}$ . Artinya, kombinasi menu optimal menurut sistem adalah sebagai berikut :

- Soto Ayam :  $\frac{1}{4}$  porsi
- Nasi : 1 porsi
- Jus Alpokat : 1 porsi

### 3. Data

#### 3.1. Data kandungan gizi bahan makanan

Data kandungan gizi bahan makanan didapat dari basis data *United States Department of Agriculture (USDA)* (U.S. Department of Agriculture, 2014).. Data dapat diakses secara bebas pada alamat <http://ndb.nal.usda.gov/ndb/>.

#### 3.2. Data satuan bahan makanan

Data satuan bahan makanan adalah data yang berisi satuan-satuan atau unit-unit dari sebuah bahan makanan. Yang dimaksud dengan satuan di sini misalnya “Sendok Makan”, “Sendok Teh” dan sebagainya. Data ini didapatkan dari sumber yang sama serta dengan metode pengumpulan yang sama dengan data kandungan gizi bahan makanan.

#### 3.3. Data menu

Data ini menjelaskan bahan-bahan pembuat, takaran masing-masing bahan, porsi, porsi penyajian standar, serta harga dari sebuah menu makanan. Situs berbagi resep [cookpad.com/id](http://cookpad.com/id) dijadikan sebagai rujukan dalam penyusunan resep-resep yang akan dijadikan data. Harga menu makanan akan dihitung berdasarkan jumlah harga bahan-bahan pembuatnya. Data mengenai porsi penyajian standar diperkirakan berdasarkan kuantitas konsumsi wajar seseorang terhadap suatu menu. Data-data ini kemudian akan dikonsultasikan dengan pakar gizi untuk mendapatkan validasi bahwa data-data tersebut valid dan layak digunakan sebagai data penelitian.

#### 3.4. Data menu set

Data ini digunakan untuk menguji optimalitas algoritma. Data berisi susunan menu-menu yang mungkin dikonsumsi seseorang dalam satu hari. Sebelum digunakan sebagai data pengujian, data menu set ini akan terlebih dahulu dikonsultasikan dengan pakar gizi untuk mendapatkan validasi bahwa kasus-kasus yang direpresentasikan dalam data memang mungkin terjadi dan data dianggap layak dijadikan sebagai data uji.

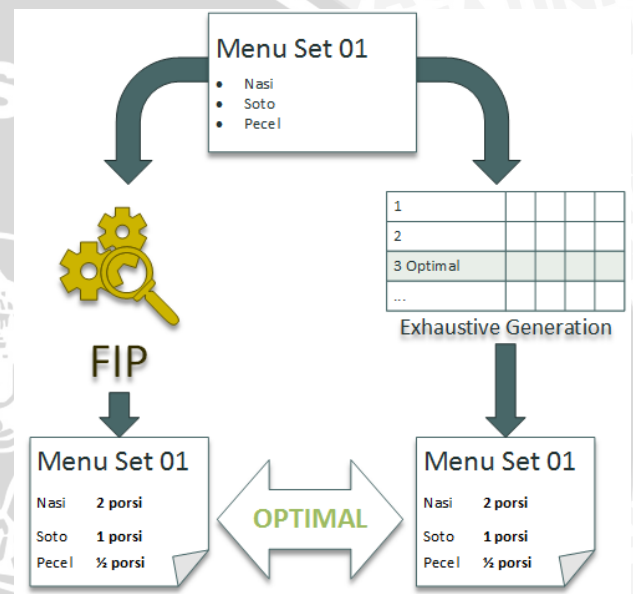
#### 3.5. Data acuan kecukupan gizi

Data ini digunakan sebagai acuan untuk menentukan kebutuhan gizi individu berdasarkan usia, jenis kelamin, berat badan, tinggi badan serta tingkat aktivitasnya. Acuan yang digunakan dalam penelitian ini adalah data DRI (*Dietary Reference Intakes*).

### 4. Strategi Pengujian

#### 4.1. Ukuran kualitas algoritma

Ukuran yang digunakan untuk menentukan kualitas algoritma FIP dalam menyelesaikan permasalahan optimasi kandungan gizi menu harian adalah tingkat optimalitasnya. Uji optimalitas dilakukan dengan membandingkan kombinasi hasil optimasi algoritma FIP dengan semua kemungkinan kombinasi yang dibangkitkan secara menyeluruh (*exhaustive*). Proses uji optimalitas dapat digambarkan dalam diagram pada Gambar 2.



Gambar 2 Diagram uji optimalitas

#### 4.2. Pembangkitan exhaustive

Untuk menguji apakah hasil optimasi dengan algoritma FIP memang merupakan kombinasi terbaik dari seluruh kombinasi yang mungkin, diperlukan sebuah strategi pembangkitan seluruh kemungkinan kombinasi. Strategi yang disebut sebagai pembangkitan *exhaustive* ini akan terlebih dahulu mendaftarkan seluruh kemungkinan kombinasi dari permasalahan yang sedang diselesaikan. Selanjutnya, setiap kombinasi akan dihitung nilai gizi dan harganya. Kombinasi-kombinasi yang tidak memenuhi kriteria (*infeasible*) kemudian disingkirkan dari daftar. Setelah daftar hanya berisi kombinasi yang memenuhi kebutuhan gizi (*feasible*), dapat diketahui kombinasi optimal, yakni kombinasi yang memiliki harga paling murah. Jika kombinasi ini sama dengan kombinasi yang diberikan oleh FIP, maka untuk kasus ini algoritma FIP dikatakan telah optimal. Tabel 2 merupakan contoh dari tabel yang akan digunakan untuk melakukan pengujian optimalitas.

Address	Energy	Protein	Vitamin A	Vitamin B1	Vitamin C	Price
	850 - Infinity	21.25 - Infinity	382.5 - Infinity	0.43 - Infinity	38.25 - Infinity	0 - 30000
0001.1.01	1,002.3 - 1,709.4 $\uparrow$ min	$\downarrow$ 15.94 - 26.48 $\uparrow$ min	1447.09 - 1,925.44 $\uparrow$ min	0.48 - 0.76 $\uparrow$ min	77.06 - 97.47 $\uparrow$ min	8,125 - 9,875 $\uparrow$ min
0001.1.10	★ 1,695.2 - 2,829.4	24.17 - 38.93	2,641.69 - 3,567.94	0.78 - 1.14	148.96 - 189.22	10,875 - 13,125
0010.1.01	1,047.7 - 1,770.8	$\downarrow$ 18.26 - 29.71	1,701.38 - 2,208.38	0.52 - 0.83	82.21 - 103.19	10,000 - 12,000
0010.1.10	1,740.6 - 2,890.8	26.49 - 42.16	2,895.38 - 3,850.88	0.81 - 1.21	154.11 - 194.94	12,750 - 15,250
0100.1.01	1,138.5 - 1,893.6	22.88 - 36.17	2,208.75 - 2,774.25	0.59 - 0.97	92.53 - 114.63	13,750 - 16,250
0100.1.10	1,831.4 - 3,013.6	31.11 - 48.62	3,402.75 - 4,416.75	0.89 - 1.35	164.43 - 206.38	16,500 - 19,500
1000.1.01	1,320.1 - 2,139.2	32.14 - 49.09	3,223.5 - 3,906	0.74 - 1.25	113.15 - 137.5	21,250 - 24,750
1000.1.10	2,013 - 3,259.2 $\uparrow$ max	40.37 - 61.54 $\uparrow$ max	4,417.5 - 5,548.5 $\uparrow$ max	1.03 - 1.63 $\uparrow$ max	185.05 - 229.25 $\uparrow$ max	24,000 - 28,000 $\uparrow$ max

- Legenda
- ★ Optimal berdasarkan FIP
  - $\uparrow$ max Nilai maksimum
  - $\uparrow$ min Nilai minimum
  - $\downarrow$  Melebihi batas bawah
  - $\uparrow$  Melebihi batas atas
  - teks Kombinasi *feasible*
  - teks Kombinasi *infeasible*
  - teks Kombinasi optimal

Tabel 2 Contoh tabel pengujian optimalitas

### 4.3. Parameter-parameter yang akan diuji

Pengujian akan dilakukan dalam empat bentuk dengan kode “Pengujian A” hingga “Pengujian D”. Masing-masing pengujian akan dilakukan dalam skenario-skenario yang bertujuan untuk mengetahui pengaruh parameter-parameter yang diuji terhadap optimalitas algoritma. Bagian ini akan menjelaskan parameter-parameter yang akan diuji.

#### 4.3.1. Keburaman berat bahan penyusun menu (menu *food weights fuzziness*)

Parameter ini menggambarkan persen toleransi atas dan bawah berat dari tiap bahan yang ada dalam suatu menu terhadap nilai tengahnya. Jika parameter ini tidak didefinisikan (atau dengan kata lain ‘bebas’), maka yang dipakai adalah toleransi asli yang didapatkan dari data. Jika parameter ini di set sebagai  $x$ , maka nilai toleransi atas dan bawah berat dari tiap bahan dalam menu akan ditimpa atau di-*override* dengan nilai sebesar  $x$  persen dari nilai tengahnya.

Nama	Bahan Utama
Sambal Goreng Ampela	- Ampela Ayam : 1 .kg <i>Fuzzy eqv. (1000, 0, 0) gram</i>
	- Wortel : 2 .buah <i>Fuzzy eqv. (122, 5, 5) gram</i>
	- Kentang : 2 .buah <i>Fuzzy eqv. (426, 10, 10) gram</i>

Tabel 3 Menu dengan menu *food weights fuzziness* = ‘bebas’

Nama	Bahan Utama
Sambal Goreng Ampela	- Ampela Ayam : 1 .kg <i>Fuzzy eqv. (1000, 0, 0) gram</i>
	- Wortel : 2 .buah <i>Fuzzy eqv. (122, 0, 0) gram</i>
	- Kentang : 2 .buah <i>Fuzzy eqv. (426, 0, 0) gram</i>

Tabel 4 Menu dengan menu *food weights fuzziness* = 0%

Nama	Bahan Utama
Sambal Goreng Ampela	- Ampela Ayam : 1 .kg <i>Fuzzy eqv. (1000, 500, 500) gram</i>
	- Wortel : 2 .buah <i>Fuzzy eqv. (122, 61, 61) gram</i>
	- Kentang : 2 .buah <i>Fuzzy eqv. (426, 213, 213) gram</i>

Tabel 5 Menu dengan menu *food weights fuzziness* = 50%

#### 4.3.2. Keburaman harga menu (menu *price fuzziness*)

Parameter ini menggambarkan persen toleransi atas dan bawah harga menu terhadap nilai tengahnya. Jika parameter ini tidak didefinisikan (atau dengan kata lain ‘bebas’), maka yang dipakai adalah toleransi asli yang didapatkan dari data. Jika parameter ini di set sebagai  $x$ , maka nilai toleransi atas dan bawah harga menu akan ditimpa atau di-*override* dengan nilai sebesar  $x$  persen dari nilai tengahnya.

Nama	Harga (s, l, r)
Sambal Goreng Ampela	(5700, 1000, 500)

Tabel 6 Harga menu dengan menu *price fuzziness* = ‘bebas’

Nama	Harga (s, l, r)
Sambal Goreng Ampela	(5700, 0, 0)

Tabel 7 Harga menu dengan menu *price fuzziness* = 0%

Nama	Harga (s, l, r)
Sambal Goreng Ampela	(5700, 2850, 2850)

Tabel 8 Harga menu dengan menu *price fuzziness* = 50%

#### 4.3.3. Keburaman batasan gizi (*reference fuzziness*)

Parameter ini menggambarkan persen toleransi terhadap batasan-batasan gizi yang telah didefinisikan. Jika parameter ini bernilai 0% artinya batasan bernilai pasti, yakni sebesar apa yang telah didefinisikan. Jika parameter ini di set sebagai  $x$  artinya batasan memiliki toleransi atas dan bawah sebesar  $x$  persen dari nilai pastinya. Parameter ini berlaku baik untuk batas-batas bawah maupun batas-batas atas. Parameter ini tidak berlaku untuk batasan harga karena harga dibatasi oleh anggaran yang dianggap pasti atau tidak memiliki toleransi.

Referensi	Lower	Upper
energy	2,507.30	Infinity
vitamin_a	900	3,000
price	0	50,000

Tabel 9 Batasan dengan *reference fuzziness* = 0%

Referensi	Lower	Upper
energy	2,256.57	Infinity
vitamin_a	810	3,300
price	0	50,000

Tabel 10 Batasan dengan *reference fuzziness* = 10%

#### 4.4. Pengujian A

Pengujian ini digunakan untuk mengetahui pengaruh keburaman data berat bahan-bahan penyusun menu yang dimasukkan oleh pengguna terhadap tingkat optimalitas algoritma. Algoritma akan diuji dengan sebanyak 50 buah menu set dalam beberapa skenario. Pada setiap skenario, harga menu dan batasan gizi diberikan sedikit keburaman, yakni sebesar 10%. Dengan kata lain, *menu price fuzziness* dan *reference fuzziness* dijadikan sebagai variabel terkontrol. Parameter *menu food weights fuzziness* sebagai variabel bebas akan diberi nilai dari 0% hingga 100% dengan kelipatan 10% tiap skenario.

Dari seluruh menu set yang dijadikan sebagai data uji, tingkat optimalitas didapatkan dari jumlah kasus yang memiliki solusi *feasible-optimal* dibandingkan dengan jumlah semua kasus yang memiliki solusi *feasible* (baik optimal maupun tak optimal).

Skenario	Keburaman Berat Bahan Penyusun Menu (%)	Keburaman Harga Menu (%)	Keburaman Batasan (%)
A.0	0	10	10
A.1	10	10	10
A.2	20	10	10
A.3	30	10	10
A.4	40	10	10
A.5	50	10	10
A.6	60	10	10
A.7	70	10	10
A.8	80	10	10
A.9	90	10	10
A.10	100	10	10

Tabel 11 Parameter dari skenario-skenario "pengujian A"

#### 4.5. Pengujian B

Pengujian ini digunakan untuk mengetahui pengaruh keburaman harga menu yang dimasukkan oleh pengguna terhadap tingkat optimalitas algoritma. Algoritma akan diuji dengan sebanyak 50 buah menu set dalam beberapa skenario. Parameter *menu price fuzziness* akan dijadikan sebagai variabel bebas yang akan diberi nilai mulai dari 0% hingga 100% dengan kelipatan 10% tiap skenario. Parameter *menu food weights fuzziness* dan *reference fuzziness* akan dijadikan sebagai variabel terkontrol dengan nilai 10%. Tingkat optimalitas didapatkan dengan cara yang sama seperti yang dilakukan pada "Pengujian A".

Skenario	Keburaman Berat Bahan Penyusun Menu (%)	Keburaman Harga Menu (%)	Keburaman Batasan (%)
B.0	10	0	10
B.1	10	10	10
B.2	10	20	10
B.3	10	30	10
B.4	10	40	10
B.5	10	50	10
B.6	10	60	10
B.7	10	70	10
B.8	10	80	10
B.9	10	90	10
B.10	10	100	10

Tabel 12 Parameter dari skenario-skenario "pengujian B"



4.6. Pengujian C

Pengujian ini digunakan untuk mengetahui pengaruh keburaman batasan-batasan gizi pada tingkat optimalitas algoritma. Pada kasus nyata, acuan kecukupan gizi memiliki nilai toleransi maksimal 10% dari nilai yang telah ditentukan. Angka 10% ini didapatkan dari hasil konsultasi dengan pakar gizi. Untuk itu, variabel bebas pengujian ini, yakni *reference fuzziness*, akan diberi nilai mulai dari 0% hingga 10% dengan kelipatan 1% tiap skenario. Parameter *menu food weights fuzziness* dan *menu price fuzziness* sebagai variabel terkontrol akan diberi nilai 10% untuk semua skenario. Tingkat optimalitas kemudian dicari dengan cara yang sama seperti yang dilakukan pada “Pengujian A”.

Skenario	Keburaman Berat Bahan Penyusun Menu (%)	Keburaman Harga Menu (%)	Keburaman Batasan (%)
C.0	10	10	0
C.1	10	10	1
C.2	10	10	2
C.3	10	10	3
C.4	10	10	4
C.5	10	10	5
C.6	10	10	6
C.7	10	10	7
C.8	10	10	8
C.9	10	10	9
C.10	10	10	10

Tabel 13 Parameter dari skenario-skenario “pengujian C”

4.7. Pengujian D

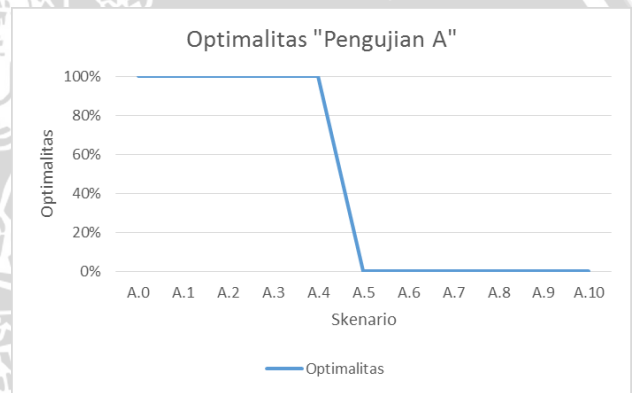
Pengujian ini digunakan untuk mengetahui optimalitas algoritma dalam kasus nyata (*practical use case*). Data yang digunakan pada “Pengujian D” ini adalah data asli yang merupakan data yang dimasukkan oleh pengguna tanpa dilakukan *override*. Artinya, parameter *menu food weights fuzziness* dan parameter *menu price fuzziness* diset sebagai ‘bebas’. Data-data ini kemudian akan diuji dengan parameter *reference fuzziness* yang diberi nilai mulai dari 0% hingga 10% dengan kelipatan 1% untuk tiap skenario.

Skenario	Keburaman Berat Bahan Penyusun Menu (%)	Keburaman Harga Menu (%)	Keburaman Batasan (%)
D.0	Bebas	Bebas	0
D.1	Bebas	Bebas	1
D.2	Bebas	Bebas	2
D.3	Bebas	Bebas	3
D.4	Bebas	Bebas	4
D.5	Bebas	Bebas	5
D.6	Bebas	Bebas	6
D.7	Bebas	Bebas	7
D.8	Bebas	Bebas	8
D.9	Bebas	Bebas	9
D.10	Bebas	Bebas	10

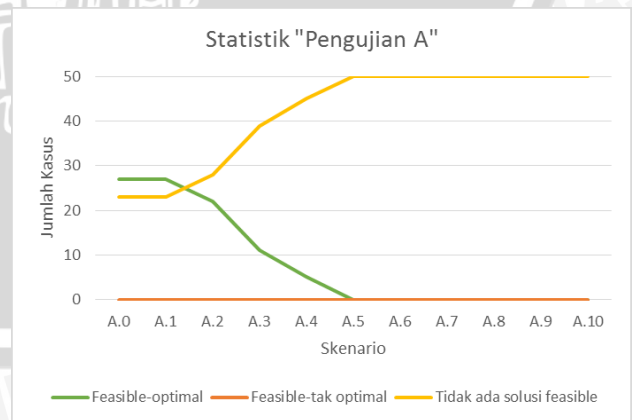
Tabel 14 Parameter dari skenario-skenario “pengujian D”

5. Hasil Pengujian

5.1. Hasil pengujian A



Gambar 3 Grafik optimalitas “pengujian A”

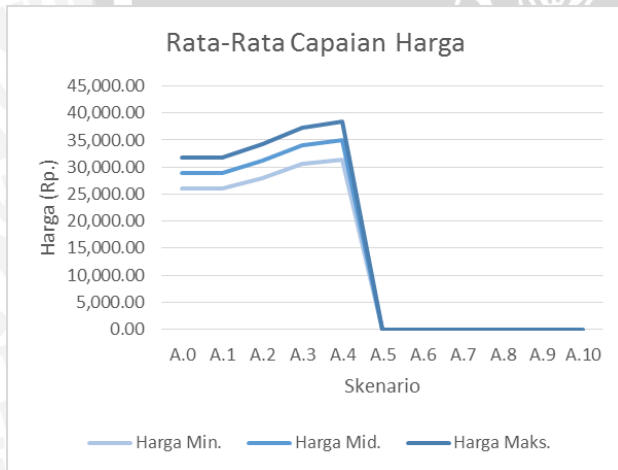


Gambar 4 Grafik statistik “pengujian A”



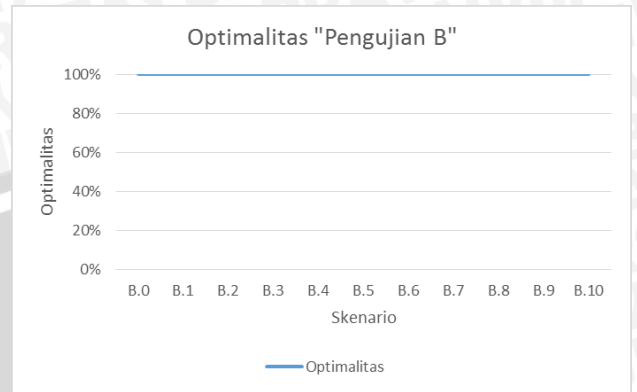
Grafik optimalitas “Penguujian A” pada Gambar 3 menunjukkan bahwa keburaman berat bahan penyusun menu tidak mempengaruhi optimalitas algoritma. Dari grafik statistik pada Gambar 4, terlihat bahwa semakin buram berat bahan penyusun menu mengakibatkan jumlah masalah yang memiliki solusi semakin menurun. Sebaliknya, jumlah masalah yang tidak memiliki solusi semakin meningkat. Hal ini menunjukkan bahwa semakin buram data berat bahan penyusun menu suatu Menu Set, maka semakin kecil kemungkinan untuk mendapatkan solusi dari Menu Set tersebut.

Penurunan jumlah solusi *feasible* ini dikarenakan dengan semakin buramnya data berat bahan penyusun menu, data kandungan nutrisi dari menu tersebut juga semakin buram. Keburaman ini mengakibatkan nilai minimum dan/atau nilai maksimum tiap nutrien untuk tiap kombinasi menu sebuah Menu Set melebar sehingga lebih mudah melewati batas bawah dan/atau batas atas yang ditentukan oleh batasan gizi. Jika setidaknya satu nutrien dari sebuah kombinasi menu melewati batasan gizi yang telah ditentukan maka kombinasi tersebut dianggap tidak memenuhi lagi untuk dijadikan sebagai kandidat solusi dan akan digantikan dengan kombinasi lain yang masih memenuhi (jika ada). Konsekuensinya adalah kombinasi pengganti tersebut akan memiliki nilai nutrisi lebih tinggi namun sekaligus harga yang lebih mahal.

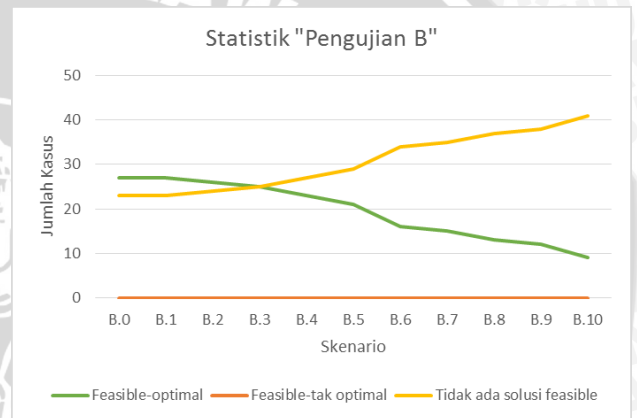


Gambar 5 Grafik rata-rata capaian harga “penguujian A”

5.2. Hasil penguujian B

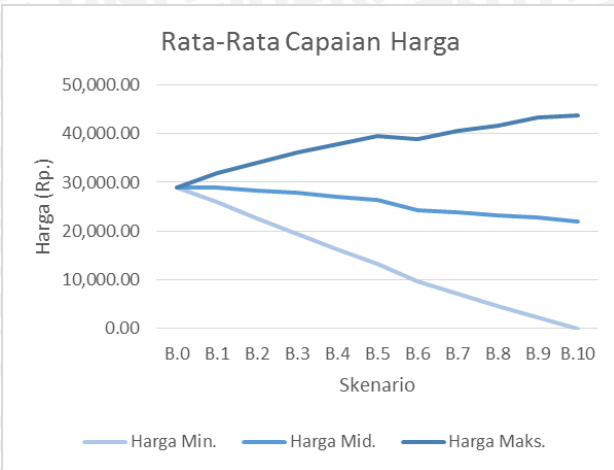


Gambar 6 Grafik optimalitas “penguujian B”



Gambar 7 Grafik statistik “penguujian B”

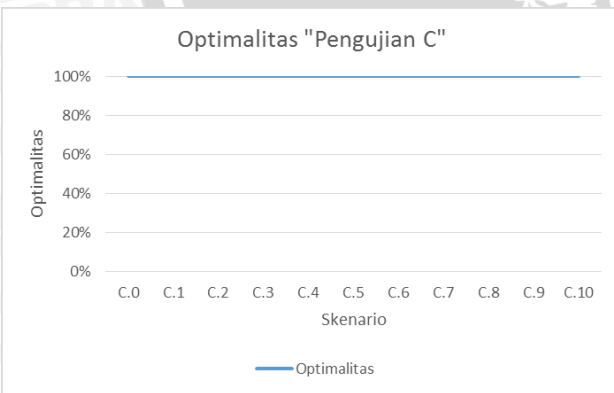
Grafik optimalitas “Penguujian B” pada Gambar 6 menunjukkan bahwa keburaman harga menu tidak mempengaruhi optimalitas algoritma. Dari grafik pada Gambar 7, terlihat bahwa semakin buram harga menu mengakibatkan jumlah masalah yang memiliki solusi semakin menurun. Efek ini sama dengan yang terjadi pada “Penguujian A”. Perbedaannya adalah bahwa pada penguujian ini yang menjadi penyebab turunnya jumlah solusi *feasible* adalah melebarnya nilai minimum dan/atau maksimum dari harga.



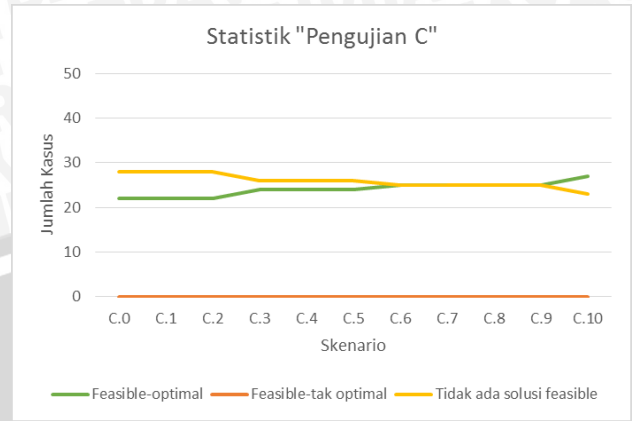
Gambar 8 Grafik rata-rata capaian harga “pengujian A”

Dari grafik pada Gambar 8 terlihat bahwa meskipun nilai minimum dan maksimum dari harga mengalami pelebaran, namun nilai tengah dari harga menunjukkan sedikit *trend* penurunan. *Trend* penurunan harga ini disebabkan oleh pelebaran nilai minimum dan/atau nilai maksimum dari harga seiring bertambahnya keburaman harga menu. Kombinasi-kombinasi dengan harga lebih tinggi nilai maksimumnya akan lebih mudah melewati batas atas daripada kombinasi-kombinasi dengan harga lebih rendah. Selain itu, jarak harga kombinasi ke batas bawah relatif lebih jauh dibandingkan dengan jarak ke batas atas. Hal ini mengakibatkan semakin banyak kombinasi-kombinasi mahal yang tereliminasi seiring dengan semakin melebarnya nilai minimum dan/atau maksimum harga dari nilai tengahnya.

5.3. Hasil pengujian C



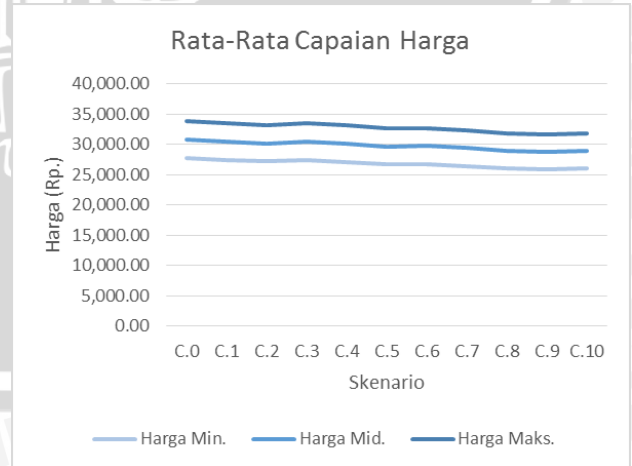
Gambar 9 Grafik optimalitas “pengujian C”



Gambar 10 Grafik statistik “pengujian C”

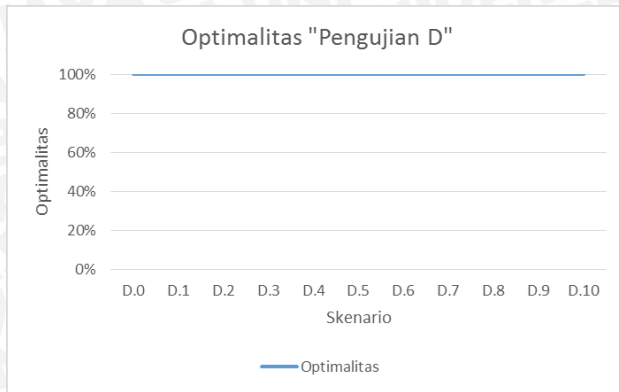
Grafik optimalitas “Pengujian C” pada Gambar 9 menunjukkan bahwa keburaman batasan gizi tidak mempengaruhi optimalitas algoritma. Dari grafik pada Gambar 10 terlihat bahwa terjadi sedikit peningkatan pada jumlah kasus dengan solusi *feasible-optimal* seiring dengan bertambahnya keburaman batasan. Artinya, semakin buram batasan, semakin besar peluang untuk mendapatkan solusi optimal. Meskipun demikian, berdasarkan panduan yang diberikan oleh pakar gizi keburaman batasan hanya diperbolehkan hingga sebesar 10%. Peningkatan jumlah kasus dengan solusi *feasible* ini dikarenakan keburaman batasan memberikan sedikit toleransi sehingga batas bawah akan menjadi lebih rendah dan batas atas menjadi lebih tinggi.

Dari pengujian ini juga didapatkan hasil bahwa semakin buram batasan, kombinasi yang dihasilkan akan semakin murah. Namun demikian, penurunan ini tidak terlalu signifikan mengingat perubahan keburaman batasan juga tidak signifikan.



Gambar 11 Grafik rata-rata capaian harga “pengujian C”

5.4. Hasil pengujian D



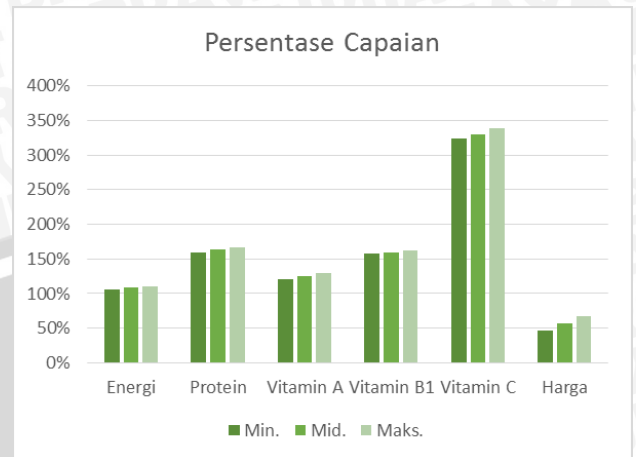
Gambar 12 Grafik optimalitas “pengujian D”

Grafik optimalitas “Pengujian D” pada Gambar 12 menunjukkan bahwa algoritma dapat diterapkan untuk kasus nyata dengan optimalitas 100%. Seperti digambarkan dalam grafik tersebut, pada kasus nyata di mana data berat bahan penyusun menu, data harga menu serta data batasan gizi memiliki keberagaman yang bervariasi, algoritma tetap dapat menemukan kombinasi paling optimal jika memang masalah tersebut memiliki solusi.

Untuk penerapan pada kasus nyata dengan keberagaman batasan sebesar 10% (skenario D.10), didapatkan persentase capaian dengan rincian seperti yang tersaji pada Tabel 15 dan dapat digambarkan dalam bentuk grafik pada Gambar 13. Dari grafik tersebut terlihat bahwa solusi-solusi yang dihasilkan oleh algoritma memiliki rata-rata nilai pemenuhan nutrisi di atas 100%. Artinya, kebutuhan nutrisi subjek dapat terpenuhi dengan baik. Dari sisi harga, terlihat pula bahwa solusi-solusi yang diberikan algoritma memiliki harga berkisar antara 46.59% hingga 66.74% dari anggaran yang diberikan. Dengan demikian, dapat dikatakan bahwa solusi-solusi yang diberikan algoritma memiliki harga yang terjangkau.

	Min.	Mid.	Maks.
Energi	105.57%	108.42%	111.03%
Protein	159.81%	163.26%	167.45%
Vitamin A	120.94%	124.59%	129.41%
Vitamin B1	157.21%	159.64%	162.97%
Vitamin C	324.39%	330.62%	339.65%
Harga	46.59%	56.75%	66.74%

Tabel 15 Persentase capaian penerapan pada kasus nyata



Gambar 13 Grafik persentase capaian penerapan pada kasus nyata

6. Kesimpulan

Berdasarkan hasil penelitian dan analisa yang telah dilakukan pada hasil penelitian, didapatkan kesimpulan-kesimpulan sebagai berikut:

1. *Fuzzy Integer Programming (FIP)* telah berhasil diterapkan sebagai algoritma untuk melakukan optimasi kandungan gizi menu harian. Penerapan ini membutuhkan data kandungan gizi bahan makanan, data satuan bahan makanan, serta data menu. Daftar menu harian yang akan dioptimasi terlebih dahulu diubah ke dalam model FIP. Model ini kemudian dikonversi ke model *Integer Programming* dengan metode konversi FIP ke MOIPP. Model hasil konversi ini kemudian diselesaikan dengan *Integer Programming*.
2. Penerapan *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian menghasilkan solusi dengan tingkat optimalitas 100%. Algoritma dapat memberikan solusi dengan rata-rata pemenuhan nutrisi di atas 100% dan dengan rata-rata penggunaan anggaran 46.59% hingga 66.74% dari anggaran yang ditentukan. Dengan demikian, dapat dikatakan bahwa solusi-solusi yang diberikan algoritma memiliki harga yang terjangkau dan memiliki nilai gizi yang mencukupi.
3. Keberagaman data masukan, berupa data berat bahan penyusun menu dan data harga menu, tidak berpengaruh pada optimalitas algoritma. Namun demikian, keberagaman tersebut memiliki pengaruh sebagai berikut:
  - a. Semakin buram berat data bahan penyusun menu mengakibatkan kemungkinan untuk mendapatkan solusi semakin kecil. Semakin buram data



berat bahan penyusun menu juga mengakibatkan solusi yang dihasilkan semakin mahal.

- b. Semakin buram harga menu mengakibatkan kemungkinan untuk mendapatkan solusi semakin kecil. Namun, semakin buram harga menu akan menghasilkan solusi yang semakin murah.
4. Keburaman batasan gizi tidak berpengaruh pada optimalitas algoritma. Semakin buram batasan, semakin besar kemungkinan untuk mendapatkan solusi. Batasan yang semakin buram juga akan menghasilkan solusi yang semakin murah. Namun, sesuai dengan panduan yang diberikan oleh pakar gizi, keburaman batasan tidak diperbolehkan melebihi angka 10%.

## 7. Saran

Berikut adalah beberapa saran yang dapat digunakan sebagai landasan pengembangan penelitian lebih lanjut:

1. Metode dalam penelitian ini dapat dikembangkan sehingga dapat diterapkan untuk orang-orang dengan diet khusus.
2. Data kandungan gizi bahan makanan serta data batasan-batasan nutrisi, jika telah tersedia, sebaiknya menggunakan data yang dikeluarkan oleh lembaga di Indonesia.
3. Diperlukan suatu data yang menjelaskan mengenai faktor penyusutan suatu bahan sehingga kandungan nutrisi suatu menu dapat dihitung secara lebih tepat.
4. Diperlukan suatu data atau formula yang menjelaskan interaksi antar nutrisi serta prioritas masing-masing nutrisi, terutama jika diterapkan pada orang-orang dengan diet khusus.
5. Penelitian ini dapat dikembangkan dengan menambahkan algoritma agar secara otomatis dapat membagi menu harian menjadi menu pagi, menu siang, menu malam serta menu selingan.

## 8. Ucapan Terima Kasih

Terima kasih kepada dosen Program Studi Ilmu Gizi Fakultas Kedokteran Universitas Brawijaya, Ibu Titis Sari Kusuma atas bantuannya dalam hal pengumpulan dan

validasi data. Terima kasih pula atas saran dan masukan-masukan yang diberikan.

## 9. Daftar Pustaka

- Dash, R. B. & Dash, P. D. P., 2012. Solving Fuzzy Integer Programming Problem as Multiobjective Integer Programming Problem. *International Journal of Fuzzy Mathematics and Systems*, Volume 2, pp. 307-314.
- Fullér, R., 1991. On product-sum of triangular fuzzy numbers. *Fuzzy Sets and Systems*, 41(1), pp. 83-87.
- Joshi, A. V., 2008. *Extension of Support Vector Machines for Imprecise Data Using Fuzzy Set Theory*. [e-book]. ProQuest. Tersedia melalui: Google Books <<https://books.google.com>> [Diakses 22 April 2016].
- Karyadi, D. & Muhilal, 1985. *Kecukupan gizi yang dianjurkan*. Jakarta: Gramedia.
- Kusuma, T. S., 2016. *Pembahasan dan Validasi Data yang Digunakan dalam Penelitian*. [diskusi] (Komunikasi personal, 13 Mei 2016).
- Oruç, K. O. et al., 2012. Menu Planning with Fuzzy 0-1 Integer Programming. *3rd International Symposium on Sustainable Development*.
- Oruç, K. O. et al., 2012. Menu Planning With Fuzzy 0-1 Integer Programming. *3rd International Symposium on Sustainable Development*.
- Prasetyo, G. A., 2010. *Aplikasi Fuzzy Linear Programming untuk Meminimalkan Biaya Pemenuhan Kebutuhan Gizi*. S1. Universitas Brawijaya.
- Prasetyo, G. A., 2010. *Aplikasi Fuzzy Linear Programming Untuk Meminimalkan Biaya Pemenuhan Kebutuhan Gizi*. S1. Universitas Brawijaya.
- Roseline, S. S. & Amirtharaj, E. H., 2012. Different strategies to solve fuzzy linear programming problems. *Recent Research in Science and Technology*, 4(5), pp. 10-14.
- Roseline, S. S. & Amirtharaj, E. H., 2012. Different Strategies to Solve Fuzzy Linear Programming Problems. *Recent Research in Science and Technology*, 4(5), pp. 10-14.
- Sufahani, S. F. & Ismail, Z., 2013. *A diet planning model for Malaysian boarding school using zero-one integer programming*.
- U.S. Department of Agriculture, A. R. S., 2014. *USDA National Nutrient Database for Standard Reference, Release 28*. [basis data] Tersedia di: <<http://ndb.nal.usda.gov/ndb/>> [Diakses 21 April 2016].

## Lampiran A. Contoh data menu makanan

### A.1. Data menu "Soto Ayam"

Nama	Soto ayam			
Porsi	4 porsi			
Harga per porsi fuzzy (s, l, r)	(8000, 500, 500)			
Porsi penyajian normal	2, 1, 1/2, 1/4			
<i>Bahan utama</i>	<i>Jumlah</i>	<i>Faktor konversi</i>	<i>Berat hasil konversi</i>	<i>Berat fuzzy (s, l, r)</i>
- Ayam	1 ekor	1 ekor = 150 gr	150 gr	(150, 10, 20)
- Kol	200 gr	1 gr = 1 gr	200 gr	(200, 50, 50)
- Bihun	50 gr	1 gr = 1 gr	50 gr	(50, 5, 5)
- Telur	3 butir	1 butir = 30 gr	90 gr	(90, 50, 10)
- Tomat	2 buah	1 buah = 100 gr	200 gr	(200, 5, 10)
<i>Bahan tambahan</i>	<i>Jumlah</i>	<i>Faktor konversi</i>	<i>Berat hasil konversi</i>	<i>Berat fuzzy (s, l, r)</i>
- Lengkuas	1 ruas jari	-	-	-
- Daun jeruk	3 lembar	-	-	-

### A.2. Data menu "Nasi"

Nama	Nasi			
Porsi	1 porsi			
Harga per porsi fuzzy (s, l, r)	(4000, 500, 500)			
Porsi penyajian normal	1			
<i>Bahan utama</i>	<i>Jumlah</i>	<i>Faktor konversi</i>	<i>Berat hasil konversi</i>	<i>Berat fuzzy (s, l, r)</i>
- Nasi	1 piring	1 piring = 200 gr	200 gr	(200, 50, 100)

### A.3. Data menu "Jus Alpokat"

Nama	Jus alpokat			
Porsi	1 porsi			
Harga per porsi fuzzy (s, l, r)	(6000, 500, 500)			
Porsi penyajian normal	1, 1/2			
<i>Bahan utama</i>	<i>Jumlah</i>	<i>Faktor konversi</i>	<i>Berat hasil konversi</i>	<i>Berat fuzzy (s, l, r)</i>
- Alpokat	3 buah	1 buah = 400 gr	1200 gr	(1200, 100, 200)
- Gula pasir	5 sendok	1 sendok = 20 gr	100 gr	(100, 50, 50)
- Susu kental manis	100 gr	1 gr = 1 gr	100 gr	(100, 20, 50)
<i>Bahan tambahan</i>	<i>Jumlah</i>	<i>Faktor konversi</i>	<i>Berat hasil konversi</i>	<i>Berat fuzzy (s, l, r)</i>
- Air Matang	250 ml	-	-	-
- Es Batu	150 gr	-	-	-

## Lampiran B. Hasil perhitungan kandungan gizi tiap menu

### B.1. Kandungan gizi menu "Soto Ayam"

Bahan	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Ayam	(113.25, 7.55, 15.1)	(6.825, 0.455, 0.91)	(9.375, 0.625, 1.25)	(303.75, 20.25, 40.5)	(0.03, 0.002, 0.004)	(0, 0, 0)
- Kol	(11, 2.75, 2.75)	(1.05, 0.2625, 0.2625)	(0.25, 0.0625, 0.0625)	(0, 0, 0)	(0.015, 0.00375, 0.00375)	(1.5, 0.375, 0.375)
- Bihun	(45, 4.5, 4.5)	(0.5875, 0.05875, 0.05875)	(0.0125, 0.00125, 0.00125)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
- Telur	(39.15, 21.75, 4.35)	(2.43, 1.35, 0.27)	(3.15, 1.75, 0.35)	(0, 0, 0)	(0.1755, 0.0975, 0.0195)	(0, 0, 0)
- Tomat	(10, 0.25, 0.5)	(0.5, 0.0125, 0.025)	(0.15, 0.00375, 0.0075)	(750, 18.75, 37.5)	(0.03, 0.00075, 0.0015)	(20, 0.5, 1)
<b>Total</b>	<b>(218.4, 36.8, 27.2)</b>	<b>(11.39, 2.14, 1.53)</b>	<b>(12.94, 2.44, 1.67)</b>	<b>(1053.75, 39, 78)</b>	<b>(0.25, 0.1, 0.03)</b>	<b>(21.5, 0.88, 1.38)</b>

### B.2. Kandungan gizi menu "Nasi"

Bahan	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Nasi	(352, 88, 176)	(7.2, 1.8, 3.6)	(0.8, 0.2, 0.4)	(0, 0, 0)	(0.2, 0.05, 0.1)	(0, 0, 0)
<b>Total</b>	<b>(352, 88, 176)</b>	<b>(7.2, 1.8, 3.6)</b>	<b>(0.8, 0.2, 0.4)</b>	<b>(0, 0, 0)</b>	<b>(0.2, 0.05, 0.1)</b>	<b>(0, 0, 0)</b>

### B.3. Kandungan gizi menu "Jus Alpokat"

Bahan	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Alpokat	(1020, 85, 170)	(10.8, 0.9, 1.8)	(78, 6.5, 13)	(2160, 180, 360)	(0.6, 0.05, 0.1)	(156, 13, 26)
- Gula Pasir	(364, 182, 182)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
- Susu Kental Mns.	(336, 67.2, 168)	(8.2, 1.64, 4.1)	(10, 2, 5)	(510, 102, 255)	(0.05, 0.01, 0.025)	(1, 0.2, 0.5)
<b>Total</b>	<b>(1720, 334.2, 520)</b>	<b>(19, 2.54, 5.9)</b>	<b>(88, 8.5, 18)</b>	<b>(2670, 282, 615)</b>	<b>(0.65, 0.06, 0.125)</b>	<b>(157, 13.2, 26.5)</b>

## Lampiran C. Persamaan-persamaan batasan FIPP

### C.1. Energi

$$2 \times (218.4, 36.8, 27.2)M_{11} + 1 \times (218.4, 36.8, 27.2)M_{12} + \frac{1}{2} \times (218.4, 36.8, 27.2)M_{13} + \frac{1}{4} \times (218.4, 36.8, 27.2)M_{14} + 1 \times (352, 88, 176)M_{21} + 1 \times (1720, 334.2, 520)M_{31} + \frac{1}{2} \times (1720, 334.2, 520)M_{32} \geq (1000, 150, 150)$$

Atau

$$(436.8, 73.6, 54.4)M_{11} + (218.4, 36.8, 27.2)M_{12} + (109.2, 18.4, 13.6)M_{13} + (54.6, 9.2, 6.8)M_{14} + (352, 88, 176)M_{21} + (1720, 334.2, 520)M_{31} + (860, 167.1, 260)M_{32} \geq (1000, 150, 150)$$

### C.2. Protein

$$(22.78, 4.28, 3.06)M_{11} + (11.39, 2.14, 1.53)M_{12} + (5.69, 1.07, 0.76)M_{13} + (2.85, 0.54, 0.38)M_{14} + (7.2, 1.8, 3.6)M_{21} + (19, 2.54, 5.9)M_{31} + (9.5, 1.27, 2.95)M_{32} \geq (25, 3.75, 3.75)$$

### C.3. Lemak

Karena acuan DRI tidak memiliki data batasan untuk lemak, maka tidak perlu membentuk batasan untuk nutrisi ini.

### C.4. Vitamin A

$$(2107.5, 78, 156)M_{11} + (1053.75, 39, 78)M_{12} + (526.87, 19.5, 39)M_{13} + (263.44, 9.75, 19.5)M_{14} + (0, 0, 0)M_{21} + (2670, 282, 615)M_{31} + (1335, 141, 307.5)M_{32} \geq (450, 67.5, 67.5)$$

*C.5. Vitamin B*

$$(0.5, 0.2, 0.06)M_{11} + (0.25, 0.1, 0.03)M_{12} + (0.125, 0.05, 0.015)M_{13} + (0.062, 0.025, 0.0075)M_{14} + (0.2, 0.05, 0.1)M_{21} + (0.65, 0.06, 0.13)M_{31} + (0.33, 0.03, 0.06)M_{32} \geq (0.5, 0.075, 0.075)$$

*C.6. Vitamin C*

$$(43, 1.76, 2.76)M_{11} + (21.5, 0.88, 1.38)M_{12} + (10.75, 0.44, 0.69)M_{13} + (5.37, 0.22, 0.35)M_{14} + (0, 0, 0)M_{21} + (157, 13.2, 26.5)M_{31} + (78.5, 6.6, 13.25)M_{32} \geq (45, 6.75, 6.75)$$

**Lampiran D. Persamaan-persamaan batasan MOIPP***D.1. Energi*

$$436.8M_{11} + 218.4M_{12} + 109.2M_{13} + 54.6M_{14} + 352M_{21} + 1720M_{31} + 860M_{32} \geq 1000$$

$$363.2M_{11} + 181.6M_{12} + 90.8M_{13} + 45.4M_{14} + 264M_{21} + 1385.8M_{31} + 692.9M_{32} \geq 850$$

$$491.2M_{11} + 245.6M_{12} + 122.8M_{13} + 61.4M_{14} + 528M_{21} + 2240M_{31} + 1120M_{32} \geq 1150$$

*D.2. Protein*

$$22.78M_{11} + 11.39M_{12} + 5.69M_{13} + 2.85M_{14} + 7.2M_{21} + 19M_{31} + 9.5M_{32} \geq 25$$

$$18.5M_{11} + 9.25M_{12} + 4.62M_{13} + 2.31M_{14} + 5.4M_{21} + 16.46M_{31} + 8.23M_{32} \geq 21.25$$

$$25.84M_{11} + 12.92M_{12} + 6.45M_{13} + 3.23M_{14} + 10.8M_{21} + 24.9M_{31} + 12.45M_{32} \geq 28.75$$

*D.3. Vitamin A*

$$2107.5M_{11} + 1053.75M_{12} + 526.87M_{13} + 263.44M_{14} + 0M_{21} + 2670M_{31} + 1335M_{32} \geq 450$$

$$2029.5M_{11} + 1014.75M_{12} + 507.37M_{13} + 253.69M_{14} + 0M_{21} + 2388M_{31} + 1194M_{32} \geq 382.5$$

$$2263.5M_{11} + 1131.75M_{12} + 565.87M_{13} + 282.94M_{14} + 0M_{21} + 3285M_{31} + 1642.5M_{32} \geq 517.5$$

*D.4. Vitamin B*

$$0.5M_{11} + 0.25M_{12} + 0.13M_{13} + 0.06M_{14} + 0.2M_{21} + 0.65M_{31} + 0.33M_{32} \geq 0.5$$

$$0.3M_{11} + 0.15M_{12} + 0.08M_{13} + 0.04M_{14} + 0.15M_{21} + 0.59M_{31} + 0.3M_{32} \geq 0.425$$

$$0.56M_{11} + 0.28M_{12} + 0.14M_{13} + 0.07M_{14} + 0.3M_{21} + 0.78M_{31} + 0.39M_{32} \geq 0.575$$

*D.5. Vitamin C*

$$43M_{11} + 21.5M_{12} + 10.75M_{13} + 5.37M_{14} + 0M_{21} + 157M_{31} + 78.5M_{32} \geq 45$$

$$41.24M_{11} + 20.62M_{12} + 10.31M_{13} + 5.15M_{14} + 0M_{21} + 143.8M_{31} + 71.9M_{32} \geq 38.25$$

$$45.76M_{11} + 22.88M_{12} + 11.44M_{13} + 5.72M_{14} + 0M_{21} + 183.5M_{31} + 91.75M_{32} \geq 51.75$$



*D.6. Anggaran*

$$16000M_{11} + 8000M_{12} + 4000M_{13} + 2000M_{14} + 4000M_{21} + 6000M_{31} + 3000M_{32} \leq 30000$$

$$15000M_{11} + 7500M_{12} + 3750M_{13} + 1875M_{14} + 3500M_{21} + 5500M_{31} + 2750M_{32} \leq 30000$$

$$17000M_{11} + 8500M_{12} + 4250M_{13} + 2125M_{14} + 4500M_{21} + 6500M_{31} + 3250M_{32} \leq 30000$$

*D.7. Batasan-batasan yang tidak melibatkan fuzzy number*

$$M_{11} + M_{12} + M_{13} + M_{14} = 1$$

$$M_{21} = 1$$

$$M_{31} + M_{32} = 1$$

$$M_{11}, M_{12}, M_{13}, M_{14}, M_{21}, M_{31}, M_{32} \geq 0, \text{ integer}$$

