Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)
Vol. 3, No. 2, Juni 2016, hlm. 133-140

OPTIMASI KANDUNGAN GIZI UNTUK MENU HARIAN MENGGUNAKAN *FUZZY INTEGER PROGRAMMING*

Lailil Muflikhah[1], Galang Gilang Ramadhan[2], Marji[3]

[1,2,3] Fakultas Ilmu Komputer, Universitas Brawijaya, Malang

e-mail: [1]lailil@ub.ac.id
(Naskah masuk: 20 Mei 2016, diterima untuk diterbitkan: 20 Juni 2016)

Abstrak

Tujuan utama dari optimasi kandungan gizi adalah membantu pengguna menemukan daftar menu harian termurah berdasarkan kebutuhan gizinya serta 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. Namun oleh karena 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 mengkonversi 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 metode apriori optimal. Metode ini diperoleh dari pembangkitan seluruh kombinasi yang mungkin sehingga 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: Kandungan gizi, Fuzzy Integer Programming, Menu Harian, Optimasi

Abstract

The main purpose of nurient optimization is to help user find the cheapest of daily menu, according to their nutrient needs and specified cost. This problem is quite challenging due to many uncertanty data are involved so that it can't be solved by using basic optimization method such as Linear Programming. Fuzzy Linear Programming is another solution that can be used. How ever, because of its characteristic is linear, this method has an effect that can produce an unpalatable food combination. Fuzzy Integer Programming (FIP) which extends Integer Programming method, apply more strictly 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 solvable easily. A test result is compared to an a priori optimal result. It is obtained from an exhaustive combination generation so that shows that this method is capable to find the optimal menu within various data fuzziness with optimality of 100%. Hence, Fuzzy Integer Programming can be considered as a reliable solution to solve optimization of daily menu nutrient.

Keywords: Nutrient, Fuzzy Integer Programming, Daily Menu, Optimization

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.

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

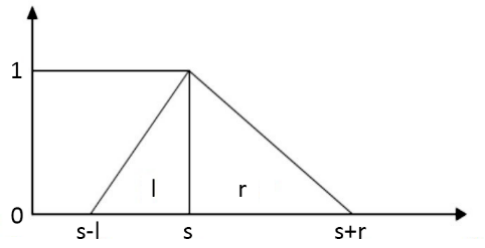
Hari Ke-	Makanan Pokok	Lauk	Sayur	Buah	Selingan
1	Nasi	Tempe Goreng	Lodeh	Pepaya	Singkong Goreng
2	Nasi	Pepes Ikan	Lalap	Pisang	Ubi Rebus
3	Nasi	Ikan Asin	Sayur Bayam	Jeruk	Jagung Rebus
4	Nasi	Tahu Goreng	Gado-Gado	Pisang	Singkong Rebus
5	Nasi	Telur Rebus	Tumis Kangkung	Pepaya	Kue Dari Singkong

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 $\tilde{A} = (a, b, c)$dan memiliki fungsi keanggotaan (Fullér, 1991):

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 - \dfrac{b - x}{b - a}, & \text{untuk } a - b \le x \le b \\ 1 - \dfrac{x - b}{c - b}, & \text{untuk } b \le x \le b + c \\ 0, & \text{untuk yang lainnya} \end{cases}$$

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):

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

  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 $x_s \leq \lfloor f \rfloor$ dan yang lain dengan menambahkan batasan $x_s \geq \lceil f \rceil$. 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 :

$$\square\square\square \sum_{\square=1}^{\square} \tilde{\square}_{\square} \square_{\square} \qquad (1)$$

Sedemikian hingga :

$$\sum_{\square \geq 0} (\square_{\square\square}, \square_{\square\square}, \square_{\square\square}) \square_{\square\square} \leq (\square_{\square}, \square_{\square}, \square_{\square}) \qquad (2a)$$

$$0 \leq \square \leq \square, 0 \leq \square \leq \square \qquad (2b)$$

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

$$\square\square\square \sum_{\square=1}^{\square} \square_{\square}(\square) \qquad (3)$$

Sedemikian hingga :

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

$$\sum_{j=1}^{n} (a_{ij} - \Delta a_{ij}) x_j \leq b_i - \Delta b_i \quad (4b)$$

$$\sum_{j=1}^{n} (a_{ij} + \Delta a_{ij}) x_j \leq b_i + \Delta b_i \quad (4c)$$

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.

1.5. 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 :

$¼ \times (150, 10, 20) = (37.5, 2.5, 5)$

- Kol :

$¼ \times (200, 50, 50) = (50, 12.5, 12.5)$

- Bihun :

$¼ \times (50, 5, 5) = (12.5, 1.25, 1.25)$

- Telur :

$¼ \times (90, 50, 10) = (22.5, 12.5, 2.5)$

- Tomat :

$¼ \times (200, 5, 10) = (50, 1.25, 2.5)$

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 nutrien 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.

1.6. Pembentukan persamaan FIPP

Tahap pertama pembentukan persamaan FIPP adalah pembentukan persamaan batasan. Misalkan terdapat Menu $M1$ dan $M2$ dengan porsi penyajian normal masing-masing $(1, ½, ¼)$ dan $(1, ¼)$, maka akan terbentuk 2 persamaan:

1) $a_{11} x11 + a_{12} x12 + a_{13} x13 + a_{14} x21 + a_{15} x22 \geq b_1$

2) $a_{21} x11 + a_{22} x12 + a_{23} x13 + a_{24} x21 + a_{25} x22 \geq b_2$

Di mana $M11$ = 1 porsi $M1$, $M12$ = ½ porsi $M1$, $M13$ = ¼ porsi $M1$, $M21$ = 1 porsi $M2$ dan $M22$ = ¼ porsi $M2$. b_1 dan b_2 adalah angka batasan nutrien atau dapat dikatakan kebutuhan gizi subjek. Diasumsikan kebutuhan gizi subjek adalah sebagai berikut:

- Energi = 1000 *Kcal*
- Protein = 25 *g/d*
- Vitamin A = 450 *µg/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 nutrien lain dapat dibentuk dan akan menghasilkan persamaan-persamaan baru. Selain batasan-batasan tersebut, diperlukan lagi beberapa batasan tambahan. Batasan tambahan pertama adalah batasan untuk menjaga total harga agar tidak melebihi anggaran:

$$(16000, 1000, 1000) x11 + (8000, 500, 500) x12 + (4000, 250, 250) x13 + (2000, 125, 125) x14 + (4000, 500, 500) x21 + (6000, 500, 500) x31 + (3000, 250, 250) x32 \leq (30000, 0, 0)$$

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

1) $x11 + x12 + x13 + x14 = 1$

2) $x21 = 1$

3) $□31 + □32 = 1$

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:

$$□11, □12, □13, □14, □21, □31, □32 \geq 0,$$
$$□□□□□□□$$

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

$$□□□. □ = 2 \times (8000,500,500)□11 +$$
$$1 \times (8000,500,500)□12 +$$
$$\frac{1}{2} \times (8000,500,500)□13 +$$
$$\frac{1}{4} \times (8000,500,500)□14 +$$
$$1 \times (4000,500,500)□21 +$$
$$1 \times (6000,500,500\ )□31 +$$
$$\frac{1}{2} \times (6000,500,500)□32$$

Atau

$$□□□. □ = (16000,1000,1000)□11 +$$
$$(8000,500,500)□12 +$$
$$(4000,250,250)□13 +$$
$$(2000,125,125)□14 +$$
$$(4000,500,500)□21 +$$
$$(6000,500,500)□31 +$$
$$(3000,250,250)□32$$

1.7. *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 :

$$□□□. □ = 18000□11 + 9000□12 +$$
$$4500□13 + 2250□14 + 5000□21 +$$
$$7000□31 + 3500□32$$

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.

1.8. *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:

$$(□11, □12, □13, □14, □21, □31, □32) =$$
$$(0,0,0,1,1,1,0)$$

Solusi dari *Integer Programming* ini juga sekaligus merupakan solusi akhir dari *Fuzzy Integer Programming*. Dari solusi tersebut diketahui bahwa menu-menu yang dipilih adalah $M14$, $M21$ dan $M31$. Artinya, kombinasi menu optimal menurut sistem adalah sebagai berikut :

- Soto Ayam : ¼ 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/ *(5)*

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 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.

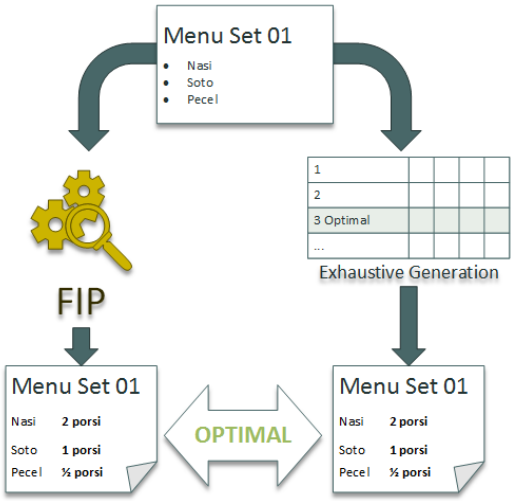
3.5 Data acuan kecukupan gizi

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 mendaftar 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.

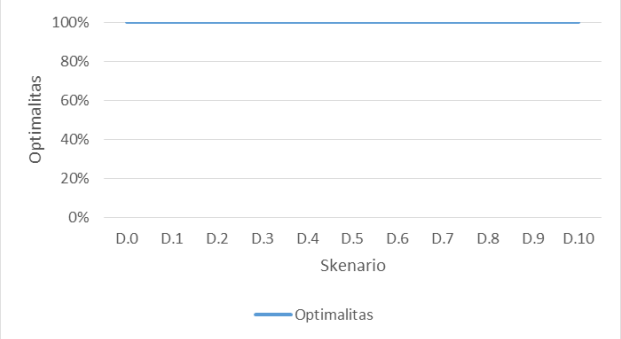
4.3 Parameter-parameter yang akan diuji

Pengujian ini digunakan untuk mengetahui optimalitas algoritma dalam kasus nyata yang bertujuan untuk mengetahui pengaruh parameter-parameter yang dixuji terhadap optimalitas *algoritma.* Data yang digunakan pada pengujian 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.

Tabel 2 Parameter dari skenario

Skenario	Keburaman Berat Bahan Penyusun	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

5. HASIL PENGUJIAN



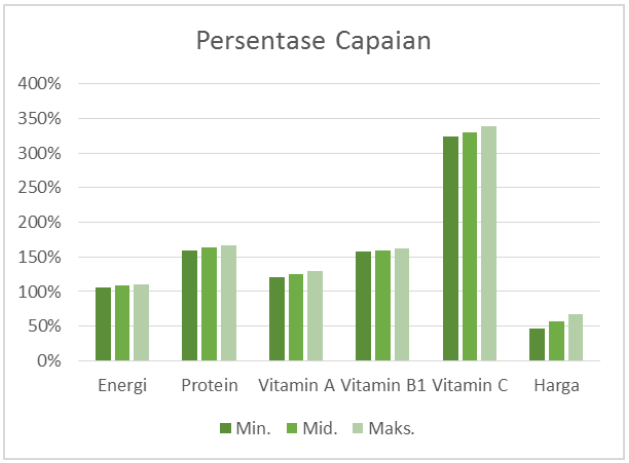
Gambar 3 Grafik optimalitas

Grafik optimalitas pada **Gambar 3** 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 keburaman yang bervariasi, algoritma tetap dapat menemukan kombinasi paling optimal jika memang masalah tersebut memiliki solusi.

Untuk penerapan pada kasus nyata dengan keburaman batasan sebesar 10% (skenario D.10), didapatkan persentase capaian dengan rincian seperti yang tersaji pada **Tabel** dan dapat digambarkan dalam bentuk grafik pada **Gambar** . 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.

Tabel 3 Persentase capaian penerapan pada kasus nyata

	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%



Gambar 4 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.

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 nutrien serta prioritas masing-masing nutrien, terutama jika diterapkan pada orang-orang dengan diet khusus.

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

Bosch, Robert dan Trick, Michael. 2014. Integer Programming

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.

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.

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.

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].