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

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Galang Gilang Ramadhan
NIM: 0910960037

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

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

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Galang Gilang Ramadhan
NIM: 0910960037

Skripsi ini telah diuji dan dinyatakan lulus pada
22 Agustus 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Lailil Muflikhah, S.Kom., M.Sc.	Drs. Marji, M.T.
NIP: 19741113 200501 2 001	NIP: 19670801 199203 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Juni 2016

Galang Gilang Ramadhan

NIM: 0910960037

KATA PENGANTAR

"We're always at the drawing board. If you're not at the drawing board, you're not making discoveries."

— Neil deGrasse Tyson

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa, yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul **"Penerapan *Fuzzy Integer Programming* untuk Optimasi Kandungan Gizi Menu Harian"**. Skripsi ini disusun sebagai salah satu syarat memperoleh gelar Sarjana Komputer pada program studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.

Tidak dapat dipungkiri bahwa terselesaikannya skripsi ini tidak lepas dari bantuan beberapa pihak. Oleh karena itu dalam kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Ayah, ibu, serta kedua adik penulis yang telah memberikan kontribusi sangat besar berupa dukungan semangat serta kasih sayang.
2. Lailil Muflikhah, S.Kom., M.Sc. selaku dosen pembimbing 1 yang telah memberikan ilmu serta masukan-masukan dengan sabar selama proses pengerjaan skripsi ini.
3. Drs. Marji, M.T. selaku dosen pembimbing 2 yang telah memberi tambahan pengetahuan serta arahan yang berkaitan dengan proses pengerjaan skripsi ini.
4. Candra Dewi, S.Kom., M.Sc. selaku dosen penasehat akademik yang telah memberikan bimbingan akademik selama penulis menjalani studi di Fakultas Ilmu Komputer.
5. Bu Titis Sari Kusuma, yang telah meluangkan waktu serta membagikan ilmu dan kepakaran beliau di bidang gizi demi membantu penulis dalam mengerjakan skripsi ini.
6. Segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama penulis menempuh pendidikan di Fakultas Ilmu Komputer.
7. Segenap staf dan karyawan Fakultas Ilmu Komputer.
8. Teman-teman Ilmu Komputer, yang baik secara langsung maupun tidak langsung, telah membantu penulis dalam menempuh pendidikan di Universitas Brawijaya.

9. Neil deGrasse Tyson, seorang astrofisikawan yang secara tidak langsung telah menginspirasi penulis dan menggugah minat penulis pada dunia ilmu pengetahuan.
10. Semua pihak yang telah berkontribusi baik secara langsung maupun tidak langsung pada skripsi ini, yang tidak dapat penulis sebutkan satu per satu. Terima kasih banyak.

Seperti makna dari kutipan yang sengaja penulis letakkan di awal kata pengantar, penulis menyadari bahwa skripsi ini masih belum dapat dikatakan sempurna. Masih banyak yang perlu diperbaiki, dikembangkan dan diterapkan lebih jauh. Oleh karena itu, kritik dan saran yang membangun akan penulis terima dengan senang hati. Penulis berharap buah pikiran yang telah penulis tuangkan dalam skripsi ini dapat bermanfaat bagi penulis sendiri, bagi pembaca serta tentunya bagi ilmu pengetahuan.

Malang, 12 Juni 2016

Penulis

galanggilangramadhan@gmail.com

repository.ub.ac.id

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 linier, 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 apriori 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*

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

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

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xii
DAFTAR GAMBAR xv
DAFTAR LAMPIRAN xvii
DAFTAR ISTILAH xviii
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 2
1.3 Tujuan 2
1.4 Manfaat 3
1.5 Batasan masalah 3
1.6 Sistematika pembahasan 3
BAB 2 LANDASAN KEPUSTAKAAN 5
2.1 Kajian pustaka 5
2.2 Diet problem 6
2.3 Menu harian 6
2.4 Acuan kecukupan gizi 7
2.5 Linear programming 9
2.5.1 *Contoh* permasalahan linear *programming* 10
2.5.2 Mendapatkan solusi dasar 11
2.5.3 Menyusun tabel simpleks awal 12
2.5.4 Perbaikan solusi 13
2.5.5 Penyusunan Table Simpleks Berikutnya 15
2.6 Fuzzy number 18
2.7 Integer programming 20

2.8 Fuzzy integer programming 25
BAB 3 METODOLOGI 28
3.1 Tipe dan strategi penelitian 28
3.2 Contoh perhitungan manual 30
3.2.1 Perhitungan nutrisi menu 31
3.2.2 Pembentukan persamaan FIPP 33
3.2.3 Konversi FIPP ke MOIPP 38
3.2.4 Pemecahan dengan *integer programming* 40
3.3 Pengumpulan data 45
3.3.1 Data kandungan gizi bahan makanan 45
3.3.2 Data satuan bahan makanan 46
3.3.3 Data menu 47
3.3.4 Data menu set 47
3.3.5 Data acuan kecukupan gizi 48
3.4 Strategi pengujian 49
3.4.1 Ukuran kualitas algoritma 49
3.4.2 Pembangkitan *exhaustive* 50
3.4.3 Parameter-parameter yang akan diuji 52
3.4.4 Pengujian pengaruh keburaman berat bahan penyusun menu terhadap optimalitas (pengujian A) 55
3.4.5 Pengujian pengaruh keburaman harga menu terhadap optimalitas (pengujian B) 55
3.4.6 Pengujian pengaruh keburaman batasan pada optimalitas (pengujian C) 55
3.4.7 Pengujian optimalitas untuk kasus nyata (pengujian D) 56
BAB 4 PERANCANGAN DAN IMPLEMENTASI 58
4.1 Arsitektur sistem 58
4.2 Spesifikasi dan kebutuhan sistem 59
4.3 Rancangan Algoritma 60
4.4 Rancangan kelas-kelas utama 62
4.4.1 Kelas Tfn *(Triangular Fuzzy Number)* 64
4.4.2 Kelas Equation dan Tuple 64
4.4.3 Kelas Nutrition 65

4.4.4 Kelas Reference 65
4.4.5 Kelas Unit 66
4.4.6 Kelas Food 66
4.4.7 Kelas StandardPortion 67
4.4.8 Kelas Menu 67
4.4.9 Kelas MenuSet 68
4.5 Perancangan basis data 68
4.5.1 Perancangan basis data untuk data menu dan bahan makanan 68
4.5.2 Perancangan basis data untuk data *reference intakes* dan *upper level* 70
4.5.3 Perancangan basis data untuk data *menu set* 71
4.5.4 Stored procedure 72
4.6 Perancangan antar muka pengguna 73
4.6.1 Antarmuka *menu set manager* 73
4.6.2 Antarmuka *solver* 74
4.6.3 Antarmuka pengujian 75
4.7 Implementasi algoritma 76
4.7.1 Fungsi *solve* 77
4.7.2 Fungsi *buildObjectiveFunction* 79
4.7.3 Fungsi *buildConstraintsFromReference* 80
4.7.4 Fungsi *buildAdditionalConstraints* 81
4.7.5 Fungsi *convertObjectiveFunction* 82
4.7.6 Fungsi *convertConstraints* 82
4.7.7 Integer programming solver 83
4.8 Implementasi antar muka pengguna 83
4.8.1 Implementasi antarmuka *menu set manager* 83
4.8.2 Implementasi antarmuka *solver* 84
4.8.3 Implementasi antarmuka pengujian 85
BAB 5 HASIL DAN PEMBAHASAN 87
5.1 Hasil pengujian 87
5.1.1 Hasil pengujian pengaruh keburaman berat bahan penyusun menu terhadap optimalitas (Pengujian A) 87

5.1.2 Hasil pengujian pengaruh keburaman harga menu terhadap optimalitas (Pengujian B) 90
5.1.3 Hasil pengujian pengaruh keburaman batasan gizi pada optimalitas (Pengujian C) 92
5.1.4 Hasil pengujian optimalitas untuk kasus nyata (Pengujian D) 95
5.2 Pembahasan 97
5.2.1 Optimalitas algoritma 97
5.2.2 Pembahasan hasil "pengujian A" 98
5.2.3 Pembahasan hasil "pengujian B" 103
5.2.4 Pembahasan hasil "Pengujian C" 107
5.2.5 Pembahasan hasil "pengujian D" 114
BAB 6 PENUTUP 117
6.1 Kesimpulan 117
6.2 Saran 118
DAFTAR PUSTAKA 119
LAMPIRAN A *Dietary Reference Intakes* – Elemen 121
LAMPIRAN B *Dietary Reference Intakes* – Nutrien Makro 122
LAMPIRAN C *Dietary Reference Intakes* – Vitamin 123
LAMPIRAN D *Upper Level* – Vitamin 124
LAMPIRAN E *Upper Level* – Elemen dan Elektrolit 125
LAMPIRAN F Data Kandungan Gizi Bahan Makanan – *Proximates* 126
LAMPIRAN G Data Kandungan Gizi Bahan Makanan – Vitamin 130
LAMPIRAN H Data Kandungan Gizi Bahan Makanan – Mineral 137
LAMPIRAN I Data Bahan Tambahan 141
LAMPIRAN J Data Satuan Bahan Makanan 143
LAMPIRAN K Data Menu 147
LAMPIRAN L Data Menu Set 161

DAFTAR TABEL

Tabel 2.1 Contoh menu harian untuk 5 hari 6
Tabel 2.2 Rumus EER berdasarkan Jenis Kelamin dan Usia 8
Tabel 2.3 Nilai PA Berdasarkan Jenis Kelamin, Usia dan Tingkat Aktivitas 9
Tabel 2.4 Kerangka tabel simpleks 12
Tabel 2.5 Kerangka konstanta pada tabel simpleks 12
Tabel 2.6 Tabel simpleks awal 13
Tabel 2.7 Tabel simpleks setelah penambahan kolom C_b dan b_i 13
Tabel 2.8 Tabel simpleks setelah penambahan baris Z_j dan C_j-Z_j 14
Tabel 2.9 Pemilihan variabel basis yang akan diganti 15
Tabel 2.10 Tabel simpleks setelah operasi baris elementer 17
Tabel 2.11 Tabel simpleks setelah penghitungan baris evaluasi netto 17
Tabel 2.12 Tabel simpleks setelah iterasi ke-2 18
Tabel 3.1 Data menu soto ayam 30
Tabel 3.2 Data menu nasi 30
Tabel 3.3 Data menu jus alpokat 31
Tabel 3.4 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (1) 32
Tabel 3.5 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (2) 32
Tabel 3.6 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (3) 32
Tabel 3.7 Perhitungan nilai *fuzzy* kandungan gizi nasi (1) 32
Tabel 3.8 Perhitungan nilai *fuzzy* kandungan gizi nasi (2) 32
Tabel 3.9 Perhitungan nilai *fuzzy* kandungan gizi nasi (3) 33
Tabel 3.10 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (1) 33
Tabel 3.11 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (2) 33
Tabel 3.12 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (3) 33
Tabel 3.13 Contoh data kandungan gizi bahan makanan 46
Tabel 3.14 Contoh data satuan bahan makanan 46
Tabel 3.15 Contoh data menu 47
Tabel 3.16 Contoh data menu set 48
Tabel 3.17 Kebutuhan gizi subyek 49
Tabel 3.18 Contoh penentuan alamat kombinasi 51

Tabel 3.19 Contoh perhitungan nilai gizi tiap kemungkinan (1) 51
Tabel 3.20 Contoh perhitungan nilai gizi tiap kemungkinan (2) 51
Tabel 3.21 Contoh tabel pengujian optimalitas 52
Tabel 3.22 Menu dengan *menu food weights fuzziness* = 'bebas' 53
Tabel 3.23 Menu dengan *menu food weights fuzziness* = 0% 53
Tabel 3.24 Menu dengan *menu food weights fuzziness* = 50% 53
Tabel 3.25 Harga menu dengan *menu price fuzziness* = 'bebas' 54
Tabel 3.26 Harga menu dengan *menu price fuzziness* = 0% 54
Tabel 3.27 Harga menu dengan *menu price fuzziness* = 50% 54
Tabel 3.28 Batasan dengan *reference fuzziness* = 0% 54
Tabel 3.29 Batasan dengan *reference fuzziness* = 10% 54
Tabel 3.30 Parameter dari skenario-skenario "pengujian A" 56
Tabel 3.31 Parameter dari skenario-skenario "pengujian B" 56
Tabel 3.32 Parameter dari skenario-skenario " pengujian C" 57
Tabel 3.33 Parameter dari skenario-skenario " pengujian D" 57
Tabel 5.1 Skenario-skenario "pengujian A" 87
Tabel 5.2 Matriks hasil "pengujian A" (1) 88
Tabel 5.3 Matriks hasil "pengujian A" (2) 88
Tabel 5.4 Statistik "pengujian A" 89
Tabel 5.5 Skenario-skenario "pengujian B" 90
Tabel 5.6 Matriks hasil "pengujian B" (1) 90
Tabel 5.7 Matriks hasil "pengujian B" (2) 91
Tabel 5.8 Statistik "pengujian B" 91
Tabel 5.9 Skenario-skenario "pengujian C" 92
Tabel 5.10 Matriks hasil "pengujian C" (1) 93
Tabel 5.11 Matriks hasil "pengujian C" (2) 93
Tabel 5.12 Statistik "pengujian C" 94
Tabel 5.13 Skenario-skenario "pengujian D" 95
Tabel 5.14 Matriks hasil "pengujian D" (1) 95
Tabel 5.15 Matriks hasil "pengujian D" (2) 96
Tabel 5.16 Statistik "pengujian D" 96
Tabel 5.17 Contoh kombinasi menu 99

Tabel 5.18 Hasil perhitungan vitamin C dari kombinasi K1 dan K2 99
Tabel 5.19 Kandungan vitamin C kombinasi K1 dan K2..... 100
Tabel 5.20 Contoh kombinasi menu 103
Tabel 5.21 Hasil perhitungan harga dari kombinasi K1 dan K2 104
Tabel 5.22 Harga kombinasi K1 dan K2..... 104
Tabel 5.23 Contoh kombinasi menu 107
Tabel 5.24 Hasil perhitungan energi dari kombinasi K1 dan K2 108
Tabel 5.25 Kandungan energi kombinasi K1 dan K2..... 108
Tabel 5.26 Data batasan energi untuk skenario C.0 hingga C.6 108
Tabel 5.27 Persentase capaian penerapan pada kasus nyata 116

DAFTAR GAMBAR

Gambar 2.1 Hubungan RDA, AI dan UL 8
Gambar 2.2 Fungsi keanggotaan *Triangular Fuzzy Number* 19
Gambar 2.3 Grafik permasalahan FIP hasil relaksasi 21
Gambar 2.4 Daerah solusi *feasible* setelah *branching* 22
Gambar 2.5 Representasi *tree* dari proses *branch-and-bound* 23
Gambar 3.1 Diagram tahapan-tahapan penelitian 28
Gambar 3.2 *Branch-and-Bound Tree* Pemecahan *Integer Programming* Iterasi-1 43
Gambar 3.3 *Branch-and-Bound Tree* Pemecahan *Integer Programming* Iterasi-2 44
Gambar 3.4 Branch-and-Bound Tree Pemecahan Integer Programming Iterasi-3 44
Gambar 3.5 *Branch-and-Bound Tree* Pemecahan Integer Programming Keseluruhan 45
Gambar 3.6 Diagram penentuan optimalitas algoritma untuk suatu Menu Set .. 50
Gambar 3.7 Ilustrasi penentuan alamat kombinasi 50
Gambar 4.1 Arsitektur aplikasi 59
Gambar 4.2 *Flowchart* algoritma 60
Gambar 4.3 *Flowchart* algoritma, subproses pembuatan model FIPP 61
Gambar 4.4 *Flowchart* algoritma, subproses konversi FIPP ke MOIPP 62
Gambar 4.5 *Class diagram* dari kelas-kelas utama (a) 62
Gambar 4.6 *Class diagram* dari kelas-kelas utama (b) 63
Gambar 4.7 Tabel-tabel yang berkenaan dengan menu dan bahan 70
Gambar 4.8 Tabel-tabel yang berkenaan dengan RI dan UL 71
Gambar 4.9 Tabel-tabel yang berkenaan dengan *menu set* 72
Gambar 4.10 Rancangan antarmuka *menu set manager* 73
Gambar 4.11 Rancangan antarmuka *solver* 74
Gambar 4.12 Rancangan antarmuka *lab* 75
Gambar 4.13 *Class diagram* FIPSolver 76
Gambar 4.14 Contoh data *menu set* 78
Gambar 4.15 Contoh data *references* 78

Gambar 4.16 Hasil implementasi antarmuka *menu set manager* 84
Gambar 4.17 Hasil implementasi antarmuka *solver* 84
Gambar 4.18 Hasil implementasi antarmuka pengujian (1) 85
Gambar 4.19 Hasil implementasi antarmuka pengujian (2) 86
Gambar 5.1 Grafik optimalitas “pengujian A” 89
Gambar 5.2 Grafik optimalitas “pengujian B” 92
Gambar 5.3 Grafik optimalitas “pengujian C” 94
Gambar 5.4 Grafik optimalitas “pengujian D” 97
Gambar 5.5 Grafik statistik “pengujian A” 98
Gambar 5.6 Grafik kandungan vitamin C kombinasi K1 dan K2 100
Gambar 5.7 Grafik rata-rata capaian “pengujian A” (1) 101
Gambar 5.8 Grafik rata-rata capaian “pengujian A” (2) 102
Gambar 5.9 Grafik statistik “pengujian B” 103
Gambar 5.10 Grafik harga kombinasi K1 dan K2 105
Gambar 5.11 Grafik rata-rata capaian “pengujian B” 106
Gambar 5.12 Grafik statistik “pengujian C” 107
Gambar 5.13 Grafik kandungan energi kombinasi K1 dan K2, skenario C.0 109
Gambar 5.14 Grafik kandungan energi kombinasi K1 dan K2, skenario C.1 109
Gambar 5.15 Grafik kandungan energi kombinasi K1 dan K2, skenario C.2 110
Gambar 5.16 Grafik kandungan energi kombinasi K1 dan K2, skenario C.3 110
Gambar 5.17 Grafik kandungan energi kombinasi K1 dan K2, skenario C.4 111
Gambar 5.18 Grafik kandungan energi kombinasi K1 dan K2, skenario C.5 111
Gambar 5.19 Grafik kandungan energi kombinasi K1 dan K2, skenario C.6 112
Gambar 5.20 Grafik rata-rata capaian “pengujian C” 113
Gambar 5.21 Grafik statistik “pengujian D” 114
Gambar 5.22 Grafik rata-rata capaian “pengujian D” 115
Gambar 5.23 Grafik persentase capaian penerapan pada kasus nyata 116

DAFTAR LAMPIRAN

LAMPIRAN A *Dietary Reference Intakes* – Elemen 121
LAMPIRAN B *Dietary Reference Intakes* – Nutrien Makro 122
LAMPIRAN C *Dietary Reference Intakes* – Vitamin 123
LAMPIRAN D *Upper Level* – Vitamin 124
LAMPIRAN E *Upper Level* – Elemen dan Elektrolit 125
LAMPIRAN F Data Kandungan Gizi Bahan Makanan – *Proximates* 126
LAMPIRAN G Data Kandungan Gizi Bahan Makanan – Vitamin 130
LAMPIRAN H Data Kandungan Gizi Bahan Makanan – Mineral 137
LAMPIRAN I Data Bahan Tambahan 141
LAMPIRAN J Data Satuan Bahan Makanan 143
LAMPIRAN K Data Menu 147
LAMPIRAN L Data Menu Set 161

DAFTAR ISTILAH

Bahan atau Bahan Makanan	Komponen dasar pembentuk makanan. Contoh bahan makanan misalnya: nasi, telur, bayam, dan lain-lain.
Menu	Sebuah 'resep' yang mendefinisikan sebuah makanan beserta komposisi bahan-bahan penyusunnya. Misalnya, menu "Nasi Goreng" mendefinisikan sebuah makanan yang tersusun atas nasi, telur, tomat, dan sebagainya.
Porsi	Dalam konteks sebuah menu, merupakan angka yang menunjukkan untuk berapa orang susunan bahan-bahan dalam menu tersebut diperuntukkan. Misalnya, menu "Nasi Goreng" memiliki porsi 2, artinya jumlah bahan-bahan yang didefinisikan dam menu tersebut diperuntukkan untuk 2 orang.
Porsi Penyajian Normal atau Porsi Normal	Dalam konteks sebuah menu, merupakan angka-angka yang menunjukkan kuantitas porsi menu yang normal dikonsumsi oleh seseorang. Misalnya, menu "Nasi Goreng" dapat memiliki porsi normal ½ porsi, 1 porsi, atau 2 porsi.
Menu Set	Sebuah daftar yang terdiri dari satu atau beberapa buah menu.
Nutrien	Unsur atau senyawa kimia yang diperlukan tubuh untuk proses metabolisme. Contoh nutrien misalnya: karbohidrat, lemak, vitamin C, dan sebagainya.
Nutrisi, Nilai Nutrisi, Nilai Gizi, atau Kandungan Gizi	Istilah yang merujuk pada nilai dari nutrien-nutrien yang terkandung dalam suatu bahan makanan atau menu.
Nilai Rujukan Nutrisi, *Nutrition Reference Value* atau *reference*	Dalam konteks nutrien, merupakan sebuah nilai rujukan yang membatasi jumlah nutrien dalam rentang angka tertentu.

Unit	Dalam konteks bahan makanan, merupakan satuan yang digunakan untuk mendefinisikan jumlah atau kuantitas tertentu dari sebuah bahan makanan. Misalnya, unit “Sendok” untuk bahan “Gula” merupakan satuan yang mendefinisikan “Gula” seberat 15 gram.
Faktor Konversi Unit	Sebuah faktor ekuivalensi sebuah unit dari sebuah bahan ke dalam satuan gram. Misalnya, unit “Sendok” untuk bahan “Gula” memiliki faktor konversi 15.
Keburaman atau *fuzziness*	Dalam konteks bilangan buram *(fuzzy number)*, menunjukkan seberapa pasti nilai dari data yang direpresentasikan dengan bilangan tersebut. *Fuzzy number* $\widetilde{A}(s,l,r) = (100,20,20)$ dikatakan lebih buram daripada $\widetilde{B}(s,l,r) = (100,10,10)$ karena data yang direpresentasikan oleh $\widetilde{A}$ dapat memiliki nilai antara 80 hingga 120, sedangkan $\widetilde{B}$ merepresentasikan data dengan nilai antara 90 hingga 110. *Fuzzy number* $\widetilde{C}(s,l,r) = (100,0,0)$ dikatakan tidak buram.

BAB 1 PENDAHULUAN

Pada bab ini, akan diberikan gambaran mengenai latar belakang penelitian, fokus permasalahan yang ingin dipecahkan, tujuan dan manfaat yang ingin dicapai serta batasan-batasan masalah yang diberlakukan. Pada akhir bab juga akan dibahas sekilas mengenai sistematika pembahasan dari laporan penelitian ini.

1.1 Latar belakang

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 bagaimana sebuah menu didefinisikan. Oruç, et al. (2012) memandang menu sebagai sebuah entitas akhir yang tidak terbagi lagi. Pada penelitian ini, sebuah menu dipandang sebagai sebuah entitas yang tersusun atas entitas-entitas lain, yakni bahan-bahan penyusunnya. Bahan-bahan dari 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, komposisi bahan akan tetap proporsional terhadap resep aslinya. Proporsionalitas komposisi bahan inilah yang dapat menjaga makanan tetap *palatable*.

1.2 Rumusan masalah

Berdasarkan permasalahan yang telah diuraikan pada bagian latar belakang, masalah yang akan diselesaikan dapat dirumuskan sebagai berikut:

1. Bagaimana menerapkan *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian?
2. Bagaimana tingkat optimalitas penerapan *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian?
3. Bagaimana pengaruh keburaman data masukkan terhadap hasil optimasi?
4. Bagaimana pengaruh keburaman batasan gizi terhadap hasil optimasi?

1.3 Tujuan

Berdasarkan identifikasi permasalahan yang telah diuraikan pada bagian rumusan masalah, maka tujuan dilakukannya penelitian ini dapat dijabarkan sebagai berikut:

1. Menerapkan metode *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian.
2. Menguji tingkat optimalitas penerapan *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian.
3. Menguji pengaruh keburaman data masukkan terhadap hasil optimasi.
4. Menguji pengaruh keburaman batasan gizi terhadap hasil optimasi.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Menyediakan solusi permasalahan optimasi gizi harian yang lebih aplikatif dan fleksibel.
2. Memberikan wawasan kepada pengguna *(end user)* bahwa susunan menu harian yang ada dapat divariasikan untuk memenuhi kebutuhan gizi harian dengan harga lebih terjangkau.

1.5 Batasan masalah

Berdasarkan uraian pada latar belakang masalah, berikut ini diberikan batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan:

1. Metode pemecahan *Fuzzy Integer Programming* yang digunakan adalah konversi FIP ke MOIPP.
2. Perhitungan kebutuhan gizi didasarkan pada kebutuhan gizi orang dewasa sehat dan tidak memiliki diet khusus.
3. Tingkat optimalitas hanya dinilai berdasarkan persentase pemenuhan nutrisi tertentu tanpa mempertimbangkan keseimbangan dan interaksi antar nutrien.
4. Perhitungan nilai gizi bahan tidak mempertimbangkan faktor penyusutan serta nilai cerna bahan.
5. Hasil optimasi adalah untuk satu hari dan independen terhadap hari lain (sisa bahan tidak dapat digunakan kembali).

1.6 Sistematika pembahasan

Laporan untuk penelitian ini akan disusun secara sistematis dan tersusun dalam enam bab dengan penjelasan sebagai berikut:

1. **BAB 1 PENDAHULUAN**

   Bab ini berisi uraian mengenai latar belakang penelitian, rumusan masalah, tujuan dan manfaat penelitian, batasan-batasan masalah penelitian serta sistematika penulisan laporan penelitian.

2. **BAB 2 LANDASAN KEPUSTAKAAN**

   Bab ini berisi pembahasan mengenai penelitian-penelitian lain yang relevan dengan topik penelitian, perbandingan, serta perbaikan-perbaikan metode yang diusulkan. Pada bab ini juga akan disertakan uraian mengenai teori-teori, konsep, dan metode-metode yang berkaitan dengan fokus penelitian.

3. **BAB 3 METODOLOGI**

   Bab ini berisi uraian mengenai metodologi dan langkah-langkah yang akan dilakukan dalam usaha untuk menerapkan dan menguji algoritma *Fuzzy Integer Programming* untuk keperluan optimasi kandungan gizi menu harian. Pembahasan meliputi tipe dan strategi penelitian, contoh perhitungan secara manual, data dan strategi pengumpulan data, serta strategi pengujian.

4. **BAB 4 PERANCANGAN DAN IMPLEMENTASI**

   Pada bab ini akan dibahas mengenai arsitektur, spesifikasi, dan kebutuhan sistem yang akan dibangun. Pembahasan dilanjutkan dengan uraian mengenai rancangan komponen-komponen utama sistem, rancangan basis data, serta rancangan antarmuka pengguna. Bab ini akan diakhiri dengan pembahasan mengenai implementasi algoritma dan implementasi antarmuka pengguna hingga dapat dihasilkan sebuah sistem yang siap uji.

5. **BAB 5 HASIL DAN PEMBAHASAN**

   Pada bab ini akan disajikan data-data hasil dari pengujian yang telah dilakukan. Dari data-data hasil pengujian yang diperoleh, akan diberikan pembahasan serta analisa-analisa terhadap hasil pengujian tersebut.

6. **BAB 6 PENUTUP**

   Bab ini berisi ringkasan mengenai hasil yang telah dicapai melalui proses penelitian berupa pernyataan-pernyataan kesimpulan disertai dengan saran-saran mengenai peluang pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan didahului dengan kajian pustaka yang akan memberikan pemahaman secara umum penelitian-penelitian sebelumnya yang berhubungan dengan penelitian ini. Kemudian, akan dilanjutkan dengan pembahasan mengenai teori, konsep, serta metode-metode yang berkaitan dengan penelitian ini.

2.1 Kajian pustaka

Ada banyak metode yang dapat digunakan untuk memecahkan masalah optimasi gizi. Metode paling sederhana adalah dengan *Linear Programming* (LP). Metode ini diawali dengan memodelkan masalah yang akan dioptimasi beserta batasan-batasannya ke dalam bentuk persamaan linier. Kemudian LP akan berusaha mencari jumlah penyajian tiap bahan makanan sehingga didapatkan komposisi bahan yang optimal. Meskipun LP memiliki beberapa keterbatasan, metode ini masih sering digunakan untuk memformulasikan diet yang sifatnya normatif (Conforti & D'Amicis, 2000).

Metode lain, yang merupakan variasi dari *Linear Programming*, adalah dengan menggunakan *Fuzzy Linear Programming* (FLP). FLP digunakan untuk memperbaiki pendekatan LP biasa dengan membuat persamaan LP lebih fleksibel terhadap data yang tidak presisi (Prasetyo, 2010). Sama dengan LP, FLP digunakan untuk menghitung jumlah penyajian tiap bahan makanan. Baik LP maupun FLP memiliki kekurangan jika digunakan untuk optimasi gizi yakni hasil optimasi dapat bernilai berapa pun asalkan tidak melanggar batasan-batasan yang ada. Pada kebanyakan kasus, hal ini menyebabkan komposisi bahan yang dihasilkan tidak enak dimakan *(unpalatable)*.

Pada penelitian oleh Sufahani dan Ismail (2013), pendekatan yang digunakan adalah dengan metode *Integer Programming (IP)*. Lebih tepatnya, metode yang digunakan disebut sebagai *Binary Integer Programming*. Penelitian ini berusaha menemukan komposisi menu optimal untuk 7 hari dari total 426 macam makanan. Untuk setiap hari, IP akan memberikan nilai 0 atau 1 bagi tiap makanan yang masih dapat dipilih (hal ini karena terdapat batasan bahwa makanan hanya boleh disajikan sekali dalam seminggu). Model yang digunakan ini disebut sebagai *non-selective menu planning*, artinya pemilihan makanan sepenuhnya dilakukan oleh sistem. Jumlah penyajian makanan telah ditentukan sebelumnya sehingga algoritma IP hanya menentukan apakah makanan tersebut disajikan atau tidak (0 atau 1). Hal inilah yang menjamin menu hasil optimasi pasti enak dimakan *(palatable)*.

Oruç, et al. (2012), dalam penelitiannya mengenai *menu planning*, menggunakan metode *Fuzzy Integer Programming* (FIP). Model perencanaan menu yang digunakan dalam penelitian tersebut juga merupakan model *non-selective*. Variabel berupa bilangan *fuzzy* digunakan untuk mengakomodasi data yang tidak presisi. Pada metode ini, hasil optimasi juga *palatable* (enak dimakan)

karena FIP hanya memilih apakah makanan disajikan atau tidak, dengan jumlah penyajian tiap makanan telah ditentukan sebelumnya.

2.2 Diet problem

Permasalahan pemenuhan kebutuhan gizi atau dikenal pula dengan *diet problem* merupakan salah satu permasalahan optimasi klasik yang mulai dipelajari pada tahun 1940-an. Berawal dari kebutuhan pemenuhan diet sehat bagi tentara Amerika, berbagai metode pemecahan *diet problem* berkembang hingga saat ini.

Menurut Matthew Ford (2006), tujuan utama pemecahan *diet problem* adalah untuk menemukan kombinasi makanan termurah yang dapat memenuhi kebutuhan nutrisi harian minimal seseorang. Selain tujuan tersebut, beberapa peneliti seperti Balintfy (1964) dan Lancaster (1992) juga memasukkan faktor *palatability* (apakah suatu makanan enak dimakan atau tidak) serta variasi (apakah suatu makanan boleh disajikan secara terus-menerus) sebagai parameter tujuan.

2.3 Menu harian

Menu harian adalah susunan menu-menu makanan yang dikonsumsi seseorang atau sebuah keluarga dalam satu hari. Idealnya, kandungan gizi dalam menu harian harus mencukupi kebutuhan gizi harian orang atau keluarga tersebut. Di Indonesia sempat dikenal slogan "4 sehat 5 sempurna" yang dapat digunakan masyarakat awam sebagai panduan untuk menyusun menu harian yang seimbang dan berkecukupan. Secara teknis, slogan ini didasarkan pada data Daftar Komposisi Bahan Makanan (DKBM). 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 2.1 diberikan contoh menu harian untuk 5 hari yang diperkirakan dapat memenuhi kebutuhan gizi harian. Data lengkap DKBM dapat dilihat pada blog milik Suyatno, Ir., MKes. (Suyatno, 2010).

Tabel 2.1 Contoh menu harian untuk 5 hari

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

Sumber: Karyadi & Muhilal (1985)

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.

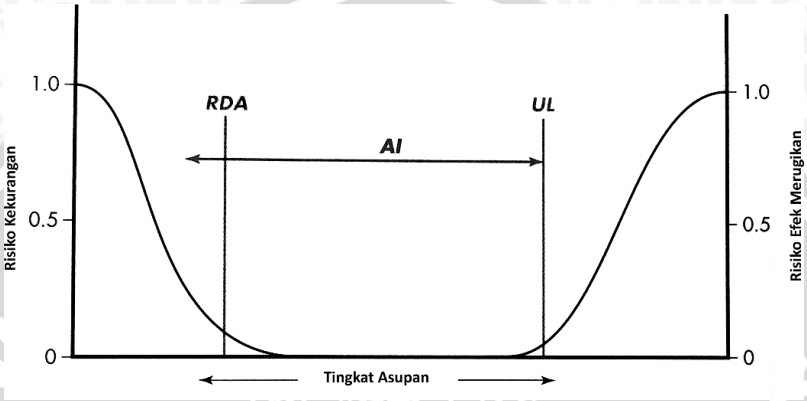
2.4 Acuan kecukupan gizi

Sebelum dapat menentukan apakah suatu perencanaan menu dapat memenuhi kebutuhan nutrisi harian seseorang atau tidak, perlu diketahui terlebih dahulu jumlah nutrisi-nutrisi esensial yang diperlukan tubuh untuk tetap dalam kondisi sehat. Atas dasar itu, beberapa standar kecukupan gizi telah ditetapkan di beberapa negara. Di Amerika dan Kanada, standar yang digunakan adalah *Dietary Reference Intakes* (DRI). Di Indonesia ditetapkan standar yang disebut dengan Angka Kecukupan Gizi (AKG).

DRI adalah standar yang dikembangkan secara kolaboratif oleh *Food and Nutrition Board* dari *Institute Of Medicine* Amerika dan Kanada (Wardlaw & Smith, 2009). DRI merupakan acuan kecukupan gizi yang melingkupi 4 standar yakni RDA, AI, UL dan EER yang dapat diuraikan sebagai berikut (Wardlaw & Smith, 2009):

- ***Recommended Dietary Allowance (RDA)***, adalah jumlah nutrisi yang direkomendasikan untuk memenuhi kebutuhan hampir semua individu (sekitar 97%) dalam kelompok umur dan jenis kelamin. Kelebihan atau kekurangan sedikit di atas atau di bawah angka RDA tidak akan menimbulkan dampak yang berarti. Namun penyimpangan sekitar 70% dari angka ini dalam jangka panjang akan mengakibatkan *deficiency* (kekurangan) atau *toxicity* (keracunan akibat kelebihan) akan nutrisi tertentu.
- ***Adequate Intake (AI)***, adalah angka pendekatan yang diasumsikan dapat memenuhi kebutuhan nutrisi seseorang untuk tetap sehat. Angka ini ditetapkan jika tidak ada informasi yang cukup untuk menentukan RDA.
- ***Tolerable Upper Intake Level*** atau ***Upper Level (UL)***, adalah jumlah asupan tertinggi sebuah nutrien sedemikian hingga tidak menimbulkan efek merugikan bagi kesehatan dalam jangka panjang. UL ditetapkan untuk beberapa vitamin dan mineral. Semakin jauh asupan melebihi nilai UL, risiko efek merugikan semakin tinggi. Gambar 2.1 menggambarkan hubungan antara RDA, AI dan UL.
- ***Estimated Energy Requirement*** **(EER)**, adalah angka yang digunakan untuk menentukan kebutuhan kalori. Angka RDA ditetapkan sedikit lebih tinggi dari kebutuhan nutrien rata-rata. Hal ini tidak masalah untuk nutrien karena kelebihan sedikit nutrien tidak berbahaya. Namun, kelebihan kalori dalam jangka panjang akan berdampak pada naiknya berat badan. Untuk itu, perhitungan kebutuhan kalori perlu lebih spesifik dengan

mempertimbangkan usia, jenis kelamin, tinggi badan, berat badan dan tingkat aktivitas fisik. Pada kasus tertentu, kebutuhan kalori tambahan untuk pertumbuhan dan laktasi juga diikutsertakan. Tabel 2.2 menyajikan rumus perhitungan EER berdasarkan jenis kelamin dan usia. Nilai koefisien *Physical Activity* (PA) berdasarkan jenis kelamin, usia, dan tingkat aktivitas disajikan pada Tabel 2.3.



Gambar 2.1 Hubungan RDA, AI dan UL

Sumber: Diadaptasi dari Wardlaw & Smith (2009)

Tabel 2.2 Rumus EER berdasarkan Jenis Kelamin dan Usia

Jenis kelamin	Usia (tahun)	*Estimated Energy Requirement* (kcal)
L	1	(89 × BB [kg] - 100) + 22
P	1	(89 × BB [kg] - 100) + 22
L	2	(89 × BB [kg] - 100) + 20
P	2	(89 × BB [kg] - 100) + 20
L	3 - 8	88.5 - (61.9 × U [thn]) + (PA × (26.7 × BB [kg] + 903 × TB [m])) + 20
P	3 - 8	135.3 - (30.8 × U [thn]) + (PA × (10.0 × BB [kg] + 934 × TB [m])) + 20
L	9 - 18	88.5 - (61.9 × U [thn]) + (PA × (26.7 × BB [kg] + 903 × TB [m])) + 25
P	9 - 18	135.3 - (30.8 × U [thn]) + (PA × (10.0 × BB [kg] + 934 × TB [m])) + 25
L	19 ke atas	662 - (9.53 × U [thn]) + (PA × (15.91 × BB [kg] + 539.6 × TB [m]))
P	19 ke atas	354 - (6.91 × U [thn]) + (PA × (9.36 × BB [kg] + 726 × TB [m]))

Keterangan:

U : Usia

BB : Berat Badan

TB : Tinggi Badan

PA : Koefisien Aktivitas Fisik (Physical Activity)

Sumber: Diadaptasi dari Wardlaw & Smith (2009)

Tabel 2.3 Nilai PA Berdasarkan Jenis Kelamin, Usia dan Tingkat Aktivitas

Jenis kelamin	Usia (tahun)	Koefisien PA *(Physical Activity)*			
		Tidak aktif	Agak aktif	Aktif	Sangat aktif
L	1	-	-	-	-
P	1	-	-	-	-
L	2	-	-	-	-
P	2	-	-	-	-
L	3 - 8	1	1.13	1.26	1.42
P	3 - 8	1	1.16	1.31	1.56
L	9 - 18	1	1.13	1.26	1.42
P	9 - 18	1	1.16	1.31	1.56
L	19 ke atas	1	1.11	1.25	1.48
P	19 ke atas	1	1.12	1.27	1.45

Sumber: Diadaptasi dari Wardlaw & Smith (2009)

AKG di Indonesia adalah acuan kecukupan gizi yang ekuivalen dengan RDA. Nilai-nilai pada AKG menunjukkan rata-rata jumlah nutrien yang dianjurkan untuk dikonsumsi oleh individu dalam suatu kelompok usia dan jenis kelamin tertentu per harinya. Nilai pada AKG hanya berlaku bagi orang sehat dan kondisi khusus yaitu ibu hamil dan ibu menyusui semester pertama dan kedua (Almatsier, 2001). AKG dimutakhirkan setiap 5 tahun sekali dengan harapan dapat memberikan andil bagi perbaikan masalah gizi dan penanggulangan penyakit tidak menular. Selain itu, Indonesia juga terus melakukan penyempurnaan pada metode perumusan AKG dengan mengkaji temuan-temuan terkini, baik di tingkat regional maupun internasional.

2.5 Linear programming

Dalam istilah matematika, *Linear Programming* (LP) adalah suatu metode yang digunakan untuk melakukan optimasi (minimasi atau maksimasi) sebuah fungsi linier dari kumpulan variabel keputusan dengan disertai beberapa batasan yang juga linier (Briend, et al., 2003). Fungsi yang akan dioptimasi disebut sebagai fungsi tujuan atau *objective function*, sedangkan batasan disebut juga dengan *constraint*. *Linear Programming* memiliki bentuk umum (Briend, et al., 2003):

Maksimumkan atau minimumkan:

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n \quad (2.1)$$

Dengan batasan:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ \leq, =, atau \ \geq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ \leq, =, atau \ \geq b_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ \leq, =, atau \ \geq b_m$$

$$x_1, x_2, \ldots, x_n \ \geq 0 \quad (2.2)$$

2.5.1 *Contoh* permasalahan linear *programming*

Pada subbab ini akan diberikan contoh kasus *Linear Programming* beserta langkah-langkah penyelesaiannya dengan menggunakan metode simpleks (Alwanta, 2012). Anggap sebuah perusahaan tas memproduksi dua jenis tas, yaitu tas model ransel dan tas model samping. Dalam proses produksinya terdapat batasan-batasan sebagai berikut:

- Untuk memproduksi 2 unit tas model ransel dan 2 unit tas model samping dibutuhkan biaya bahan baku hingga 800 (dalam ribu rupiah).
- Untuk memproduksi 2 unit tas ransel dan 2,3 unit tas samping dibutuhkan biaya hingga 1000 (dalam ribu rupiah) untuk penjahitan.
- Untuk menyelesaikan satu unit tas ransel dan 0,5 unit tas samping dibutuhkan biaya hingga 300 (dalam ribu rupiah) untuk penyelesaian.
- Untuk memproduksi 2 unit tas ransel dan 1,5 unit tas samping dibutuhkan biaya 650 (dalam ribu rupiah) untuk pemeriksaan dan pengepakan.

Perusahaan ingin mengetahui berapakah biaya maksimal yang harus dikeluarkan perusahaan tersebut dalam setiap produksi 3 unit tas ransel dan 2 unit tas samping.

Jika tas model ransel dimisalkan sebagai X_1 dan tas model samping dimisalkan sebagai X_2, maka permasalahan dapat dituliskan ke dalam model matematika sebagai berikut:

$$Maks.Z = 3X_1 + 2X_2 \quad (2.3)$$

Dengan batasan:

$$2X_1 + 2X_2 \leq 800 \quad (2.4a)$$

$$2X_1 + 2.3X_2 \leq 1000 \quad (2.4b)$$

$$X_1 + 0.5X_2 \leq 300 \quad (2.4c)$$

$$2X_1 + 1.5X_2 \leq 650 \quad (2.4d)$$

Selanjutnya, permasalahan perlu diubah ke dalam bentuk standar dengan menambahkan apa yang disebut sebagai variabel *Slack* sebagai berikut:

$$Maks.Z = 3X_1 + 2X_2 + 0S_1 + 0S_2 + 0S_3 + 0S_4 \quad (2.5)$$

Dengan batasan:

$$2X_1 + 2X_2 + S_1 = 800 \quad (2.6a)$$

$$2X_1 + 2.3X_2 + S_2 = 1000 \quad (2.6b)$$

$$X_1 + 0.5X_2 + S_3 = 300 \quad (2.6c)$$

$$2X_1 + 1.5X_2 + S_4 = 650 \quad (2.6d)$$

$$X_1, X_2, S_1, S_2, S_3, S_4 \geq 0 \quad (2.6e)$$

Setelah ditambahkannya variabel *Slack*, keempat fungsi pembatas tersebut (Persamaan 2.6a hingga Persamaan 2.6d) menjadi sistem persamaan dengan enam variabel. Apabila suatu sistem persamaan memiliki jumlah variabel lebih banyak dibanding dengan jumlah persamaannya, maka solusi persamaan tersebut adalah *infinity*. Untuk itu diperlukan metode simpleks untuk mencari solusi terbaik. Dalam prosesnya, metode simpleks akan mengeliminasi solusi-solusi yang tidak memenuhi persyaratan non-negatif.

2.5.2 Mendapatkan solusi dasar

Sistem persamaan yang telah terbentuk dari permasalahan sebelumnya memiliki jumlah variabel lebih banyak dibanding jumlah persamaannya, maka metode simpleks akan memberi nilai nol untuk dua variabel dan mencari solusi terbaik bagi empat variabel lainnya. Misalkan $X_2 = 0$ dan $S_1 = 0$, sehingga persamaan tersebut menjadi:

$$2X_1 = 800 \quad (2.7a)$$

$$2X_1 + S_2 = 1000 \quad (2.7b)$$

$$X_1 + S_3 = 300 \quad (2.7c)$$

$$2X_1 + S_4 = 650 \quad (2.7d)$$

Dengan ditetapkannya X_2 dan S_1 sebagai variabel bernilai 0, maka sistem persamaan tersebut telah menjadi sistem persamaan empat variabel (X_1, S_2, S_3, S_4). Dengan demikian nilai dari variabel-variabel tersebut dapat dicari:

- Dari Persamaan (2.7a) diperoleh $X_1 = \frac{800}{2} = 400$.
- Dari Persamaan (2.7b) diperoleh $S_2 = 1000 - (2 \times 400) = 200$.
- Dari Persamaan (2.7c) diperoleh $S_3 = 300 - 400 = -100$.
- Dari Persamaan (2.7d) diperoleh $S_4 = 650 - (2 \times 400) = -150$.

Dengan demikian, solusi dari permasalahan tersebut adalah:

$$X_1 = 400$$
$$X_2 = 0$$
$$S_1 = 0$$
$$S_2 = 200$$
$$S_3 = -100$$
$$S_4 = -150 \quad (2.8)$$

Solusi ini disebut sebagai solusi dasar. Prosedur umum untuk mendapatkan solusi dasar adalah dengan membangun bentuk persamaan standar untuk n variabel (termasuk variabel keputusan dan *Slack*) dan m persamaan pembatas, di mana n lebih besar dari m. Selanjutnya, tetapkan $n - m$ variabel mana saja

sebagai variabel bukan basis dan beri nilai nol. Kemudian, temukan solusi dari m persamaan pembatas untuk m variabel lainnya. Pada permasalahan perusahaan tas, karena solusi yang diperoleh untuk S_3 dan S_4 masih bernilai negatif, maka solusi ini masih belum merupakan solusi yang memenuhi atau solusi *feasible*. Oleh karena itu pemilihan variabel basis perlu diubah.

2.5.3 Menyusun tabel simpleks awal

Tahap pertama dalam menyelesaikan permasalahan *Linear Programming* menggunakan metode simpleks adalah membangun tabel simpleks awal. Tabel 2.4 menggambarkan kerangka dari tabel simpleks.

Tabel 2.4 Kerangka tabel simpleks

Baris C	
Matriks A	**Kolom B**

Sumber: Alwanta (2012)

Isi dari masing-masing bagian dijelaskan sebagai berikut:

- Baris C = baris dari koefisien fungsi tujuan.
- Kolom B = kolom dari nilai sisi kanan masing-masing batasan.
- Matriks A = matriks $m \times n$ dari koefisien masing-masing variabel pada setiap batasan.

Dari kerangka tersebut, tabel simpleks awal dibangun dengan memberikan notasi pada semua konstanta sehingga terbentuk susunan sebagaimana digambarkan pada Tabel 2.5.

Tabel 2.5 Kerangka konstanta pada tabel simpleks

C_1	C_2	...	C_n	b_i
a_{11}	a_{12}	...	a_{1n}	b_1
a_{21}	a_{22}	...	a_{2n}	b_2
.	.	...	.	...
a_{m1}	a_{m2}	...	a_{mn}	b_m

Sumber: Alwanta (2012)

Di mana:

C_j = konstanta fungsi tujuan untuk variabel *j*

b_i = konstanta sisi kanan untuk batasan ke-*i*

a_{ij} = konstanta yang berasosiasi dengan variabel *j* pada batasan ke-*i*

Permasalahan pada perusahaan tas yang dibahas sebelumnya dapat ditransformasikan ke dalam bentuk tabel simpleks dengan bentuk awal seperti pada Tabel 2.6.

Tabel 2.6 Tabel simpleks awal

X1	X2	S1	S2	S3	S4	
3	**2**	**0**	**0**	**0**	**0**	**b_i**
2	2	1	0	0	0	800
2	2.3	0	1	0	0	1000
1	0.5	0	0	1	0	300
2	1.5	0	0	0	1	650

Sumber: Alwanta (2012)

Dari tabel simpleks awal (Tabel 2.6) dapat diketahui dengan mudah solusi *feasible* dasar awal (*initial basic feasible solution*) karena untuk setiap kolom variabel basis terdapat koefisien yang bernilai 1, demikian juga untuk setiap barisnya. Nilai dari variabel basis dengan demikian dinyatakan oleh nilai pada kolom b_i, misalnya S_2 bernilai 1000.

2.5.4 Perbaikan solusi

Untuk mengoptimalkan solusi, metode simpleks harus mencari solusi *feasible* dasar baru yang dapat menghasilkan perbaikan pada nilai fungsi tujuan. Hal ini dilakukan dengan cara mengganti salah satu variabel basis dengan variabel non-basis. Metode simpleks memberikan cara dan prosedur untuk penggantian variabel ini. Untuk itu, diperlukan dua kolom tambahan pada bagian kiri tabel simpleks awal. Satu kolom diberi label $Basis$ dan satu kolom lain diberi label C_b. Kolom $Basis$ berisi nama variabel basis, sedangkan kolom C_b berisi nilai koefisien variabel tersebut seperti terdapat pada fungsi tujuan. Dengan demikian, terbentuk tabel simpleks seperti pada Tabel 2.7.

Tabel 2.7 Tabel simpleks setelah penambahan kolom C_b dan b_i

		X1	X2	S1	S2	S3	S4	
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**
S1	0	2	2	1	0	0	0	800
S2	0	2	2.3	0	1	0	0	1000
S3	0	1	0.5	0	0	1	0	300
S4	0	2	1.5	0	0	0	1	650

Sumber: Alwanta (2012)

Selain itu, perlu ditambahkan dua baris lagi di bagian bawah tabel simpleks yakni baris Z_j dan baris $C_j - Z_j$. Baris Z_j menunjukkan perubahan pada nilai fungsi tujuan jika satu unit variabel matriks *A* pada kolom *j* dimasukkan menjadi variabel

basis. Baris $C_j - Z_j$ atau disebut pula baris evaluasi *netto* menunjukkan pengaruh *netto* dari masuknya variabel baru tersebut pada nilai fungsi tujuan.

Kolom C_b merupakan koefisien dari variabel dasar pada fungsi tujuan. Untuk menghitung perubahan nilai fungsi tujuan (Z_j) apabila nilai variabel dasar X_j meningkat dari nol menjadi 1 perhitungannya adalah:

$$Z_1 = 0(2) + 0(2) + 0(1) + 0(2) = 0$$
$$Z_2 = 0(2) + 0(2.3) + 0(0.5) + 0(1.5) = 0$$
$$Z_3 = 0(1) + 0(0) + 0(0) + 0(0) = 0$$
$$Z_4 = 0(0) + 0(1) + 0(0) + 0(0) = 0$$
$$Z_5 = 0(0) + 0(0) + 0(1) + 0(0) = 0$$
$$Z_6 = 0(0) + 0(0) + 0(1) + 0(1) = 0 \quad (2.9)$$

Sampai pada tahap ini, nilai koefisien C_1, C_2, C_3, C_4, C_5 dan C_6 pada fungsi tujuan masing-masing bernilai 3, 2, 0, 0, 0 dan 0. Maka nilai $C_1 - Z_1 = 3$, $C_2 - Z_2 = 2$ dan seterusnya. Apabila angka-angka tersebut dimasukkan ke dalam tabel simpleks, maka akan terbentuk Tabel 2.8. Pada kolom b_i bagian bawah diperoleh nilai $Z_j = 0$ yang berasal dari perkalian antara kolom b_i dengan koefisien variabel basisnya.

Tabel 2.8 Tabel simpleks setelah penambahan baris Z_j dan C_j-Z_j

		X1	X2	S1	S2	S3	S4	
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**
S1	0	2	2	1	0	0	0	800
S2	0	2	2.3	0	1	0	0	1000
S3	0	1	0.5	0	0	1	0	300
S4	0	2	1.5	0	0	0	1	650
	Z_j	0	0	0	0	0	0	0
	C_j-Z_j	3	2	0	0	0	0	

Sumber: Alwanta (2012)

Variabel X_1 memiliki nilai netto positif terbesar, yakni 3, sehingga variabel ini dimasukkan menggantikan salah satu variabel basis yang ada. Untuk memilih variabel basis mana yang akan diganti maka dicari variabel yang berasosiasi dengan batasan yang paling membatasi nilai X_1. Untuk mengetahui batasan mana yang paling membatasi X_1 perlu dihitung rasio antara nilai b_i dibandingkan dengan nilai X_1 pada *constraint* tersebut. Misalnya untuk batasan pertama, rasio yang didapatkan adalah $800/2 = 400$. Setelah rasio tiap batasan diketahui, dipilih batasan dengan rasio terkecil. Pada kasus ini batasan yang paling membatasi X_1 adalah batasan ketiga karena tidak mengizinkan nilai X_1 melebihi 300. Oleh karena itu variabel yang bersesuaian dengan batasan ini, yakni S_3, digantikan atau dengan kata lain berubah dari basis menjadi non-basis. Tabel 2.9 mengilustrasikan proses pemilihan variabel yang akan diganti.

Tabel 2.9 Pemilihan variabel basis yang akan diganti

		X1	X2	S1	S2	S3	S4		
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**	**b_i / a_{i1}**
S1	0	2	2	1	0	0	0	800	800 / 2 = 400
S2	0	2	2.3	0	1	0	0	1000	1000 / 2 = 500
S3	0	1	0.5	0	0	1	0	300	300 / 1 = 300
S4	0	2	1.5	0	0	0	1	650	650 / 2 = 325
	Z_j	0	0	0	0	0	0	0	
	C_j-Z_j	3	2	0	0	0	0		

Sumber: Alwanta (2012)

Elemen matriks yang merupakan perpotongan antara kolom X_1 dengan baris S_3 disebut dengan elemen *pivot*, sedangkan baris dan kolom yang berasosiasi dengan elemen pivot disebut baris *pivot* dan kolom *pivot*.

2.5.5 Penyusunan Table Simpleks Berikutnya

Untuk menemukan solusi yang baru setelah masuknya variabel X_1 pada permasalahan sebelumnya maka perlu dilakukan perubahan pada tabel simpleks terakhir. Masuknya variabel X_1 harus tetap menjaga pola kolom variabel yang digantikannya, yakni variabel S_3 dengan pola:

$$\begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}$$

Prosedur transformasi tabel simpleks sehingga tetap menghasilkan sistem persamaan batasan yang ekuivalen adalah dengan mengikuti langkah-langkah "Operasi Baris Elementer". Operasi baris elementer tidak mengubah solusi dari sistem persamaan tetapi mengubah koefisien dari variabel dan nilai sisi kanan batasan. Tujuan operasi baris elementer adalah untuk mentransformasi sistem persamaan batasan menjadi bentuk yang mudah untuk diidentifikasi solusi basis barunya. Operasi baris elementer dilakukan dengan cara:

1. Kalikan setiap baris (persamaan) dengan bilangan bukan nol
2. Gantikan setiap baris (persamaan) dengan hasil penambahan atau pengurangan berganda dari baris lainnya.

Kembali ke tabel simpleks terakhir, kolom X_1 yang sebelumnya:

$$a_{11} = 2$$
$$a_{21} = 2$$
$$a_{31} = 1$$
$$a_{41} = 2$$

Akan ditransformasi sehingga menyerupai pola kolom variabel S_3 menjadi:

$$a_{11} = 0$$
$$a_{21} = 0$$
$$a_{31} = 1$$
$$a_{41} = 0$$

Dengan langkah-langkah sebagai berikut:

- Elemen matriks a_{31} sudah bernilai 1 sehingga tidak diperlukan operasi baris elementer pada persamaan baris ketiga. Baris ini sekaligus menjadi baris *pivot* karena berkaitan dengan *constraint* yang paling membatasi X_1.
- Untuk membuat elemen a_{11} menjadi 0 operasi baris elementer yang dilakukan adalah dengan mengalikan baris *pivot* dengan 2 sehingga menjadi:

  $$2(1X_1 + 0.5X_2 + 0S_1 + 0S_2 + 1S_3 + 0S_4) = 2(300)$$

  Atau:

  $$2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4 \quad = 600$$

  Selanjutnya, dilakukan pengurangan baris pertama dengan persamaan baris *pivot* menjadi:

  $$(2X_1 + 2X_2 + 1S_1) - (2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4)$$
  $$= 800 - 600$$

  Atau:

  $$0X_1 + 1X_2 - 1S_1 - 0S_2 - 2S_3 - 0S_4 \quad = 200 \qquad (2.10)$$

- Untuk membuat elemen a_{21} menjadi 0 operasi yang dilakukan adalah mengalikan baris *pivot* dengan 2 sehingga menjadi:

  $$2(1X_1 + 0.5X_2 + 0S_1 + 0S_2 + 1S_3 + 0S_4) = 2(300)$$

  Atau:

  $$2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4 = 600$$

  Kemudian dilakukan pengurangan terhadap persamaan baris kedua sehingga baris kedua menjadi:

  $$(2X_1 + 3.3X_2 + 1S_2) - (2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4)$$
  $$= 1000 - 600$$

  Atau:

  $$0X_1 + 2.3X_2 - 0S_1 + 1S_2 - 2S_3 - 0S_4 \; = 400 \qquad (2.12)$$

- Untuk membuat elemen a_{41} menjadi 0 maka baris *pivot* juga harus dikalikan dengan 2 sehingga menjadi:

  $$2(1X_1 + 0.5X_2 + 0S_1 + 0S_2 + 1S_3 + 0S_4) = 2(300) \qquad (2.13)$$

Atau:

$$2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4 = 600$$

Kemudian, lakukan pengurangan terhadap persamaan baris keempat sehingga menjadi:

$$(2X_1 + 1.5X_2 + 1S_4) - (2X_1 + 1X_2 + 0S_1 + 0S_2 + 2S_3 + 0S_4)$$

$$= 650 - 600$$

Atau:

$$0X_1 + 0.5X_2 - 0S_1 - 1S_2 - 2S_3 + 1S_4 = 50$$

Setelah dilakukan operasi baris elementer, maka tabel simpleks yang terbentuk saat ini adalah seperti pada Tabel 2.10.

Tabel 2.10 Tabel simpleks setelah operasi baris elementer

		X1	X2	S1	S2	S3	S4	
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**
S1	0	0	1	1	0	-2	0	200
S2	0	0	2.3	0	1	-2	0	400
X1	3	1	0.5	0	0	1	0	300
S4	0	0	0.5	0	0	-2	1	50

Sumber: Alwanta (2012)

Tahap berikutnya adalah menghitung nilai Z_j dan $C_j - Z_j$ untuk masing-masing kolom, termasuk nilai Z_j pada kolom b_i. Setelah tahap tersebut dilakukan, akan terbentuk tabel simpleks seperti pada Tabel 2.11.

Tabel 2.11 Tabel simpleks setelah penghitungan baris evaluasi netto

		X1	X2	S1	S2	S3	S4	
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**
S1	0	0	1	1	0	-2	0	200
S2	0	0	2.3	0	1	-2	0	400
X1	3	1	0.5	0	0	1	0	300
S4	0	0	0.5	0	0	-2	1	50
	Z_j	3	1.5	0	0	3	0	900
	C_j-Z_j	0	0.5	0	0	-3	0	

Sumber: Alwanta (2012)

Proses selanjutnya adalah terus mengusahakan perbaikan pada nilai fungsi tujuan dengan memasukkan variabel non-basis menjadi variabel basis baru seperti langkah-langkah pada iterasi pertama. Dari evaluasi baris netto diketahui bahwa variabel X_2 memiliki nilai positif terbesar (0.5) sehingga ia menjadi variabel basis baru. Kolom X_2 dipilih menjadi kolom *pivot*. Untuk memilih baris *pivot* dipilih baris

dengan nilai rasio (b_i/a_{i2}) terkecil. Dalam hal ini batasan ke-4 adalah yang paling membatasi sehingga variabel S_4 saat ini menjadi non-basis.

Setelah dilakukan operasi baris elementer diikuti dengan penghitungan nilai Z_j dan $C_j - Z_j$ seperti pada iterasi pertama, iterasi kedua ini menghasilkan tabel simpleks seperti pada Tabel 2.12.

Tabel 2.12 Tabel simpleks setelah iterasi ke-2

		X1	X2	S1	S2	S3	S4	
Basis	**C_b**	**3**	**2**	**0**	**0**	**0**	**0**	**b_i**
S1	0	0	0	1	0	2	-2	100
S2	0	0	0	0	1	7.2	-4.6	170
X1	3	1	0	0	0	3	-1	250
X2	2	0	1	0	0	-4	2	100
	Z_j	3	2	0	0	1	1	950
	C_j-Z_j	0	0	0	0	-1	-1	

Sumber: Alwanta (2012)

Dari evaluasi baris netto tidak ditemukan lagi nilai positif yang berarti solusi yang dihasilkan sudah maksimum, yaitu $X_1 = 250$ dan $X_2 = 100$ serta menghasilkan nilai fungsi tujuan 950 (dalam ribu rupiah).

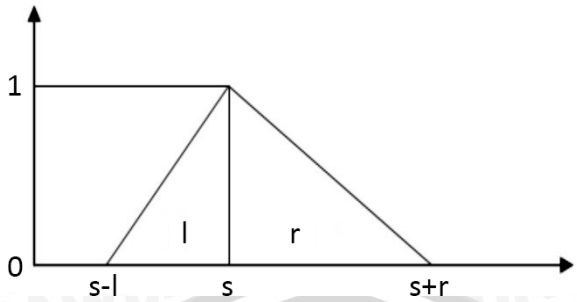
2.6 Fuzzy number

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.

Di antara bermacam-macam tipe *fuzzy number*, *triangular* dan *trapezoidal fuzzy number* termasuk yang paling penting (Roseline & Amirtharaj, 2012). Penelitian ini akan menggunakan *triangular fuzzy number*, sehingga pembahasan akan dibatasi pada tipe tersebut. *Triangular fuzzy number* A, dinotasikan sebagai $\tilde{A} = (s, l, r)$, memiliki fungsi keanggotaan (Fullér, 1991):

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 - \dfrac{s-x}{l}, & untuk\ s - l \le x \le s \\ 1 - \dfrac{x-s}{r}, & untuk\ s \le x \le s + r \\ 0, & untuk\ yang\ lain \end{cases} \quad (2.14)$$

s adalah titik tertinggi, sedangkan $l > 0$ adalah *left spread* dan $r > 0$ adalah *right spread*. Jika $l = r$ maka *triangular fuzzy number* tersebut disebut *symmetric triangular fuzzy number*. Gambar 2.2 menggambarkan grafik fungsi keanggotaan dari sebuah *Triangular Fuzzy Number*.



Gambar 2.2 Fungsi keanggotaan *Triangular Fuzzy Number*

Sumber: Diadaptasi dari Roseline & Amirtharaj (2012)

Pada $s - l$ dan $s + r$ nilai fungsi keanggotaannya adalah 0, artinya data yang direpresentasikan dengan bilangan ini nilainya tidak mungkin lebih rendah dari $s - l$ atau lebih tinggi dari $s + r$. Pada s, nilai fungsi keanggotaannya adalah 1, artinya data yang direpresentasikan dengan bilangan ini 'biasanya' atau 'kemungkinan besar' bernilai sebesar s.

Seperti halnya pada bilangan biasa, pada bilangan *fuzzy* juga dapat dilakukan operasi-operasi matematis. 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):

- Penjumlahan

  $$\tilde{A} + \tilde{B} = (s_1, l_1, r_1) + (s_2, l_2, r_2) = (s_1 + s_2, l_1 + l_2, r_1 + r_2) \quad (2.15)$$

- Pengurangan

  $$\tilde{A} - \tilde{B} = (s_1, l_1, r_1) - (s_2, l_2, r_2) = (s_1 - s_2, l_1 - l_2, r_1 - r_2) \quad (2.16)$$

- Perkalian skalar

  $$x > 0, \quad \tilde{A}x = (s_1 x, l_1 x, r_1 x) \quad (2.17a)$$

  $$x < 0, \quad \tilde{A}x = (r_1 x, l_1 x, s_1 x) \quad (2.17b)$$

- Pengurutan parsial

  $A \leq B$ 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:

  $$s_1 \leq s_2 \quad (2.18a)$$

  $$s_1 - l_1 \leq s_2 - l_2 \quad (2.18b)$$

  $$s_1 + r_1 \leq s_2 + l_2 \quad (2.18c)$$

2.7 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* dikategorikan lagi menjadi:

- **MIP *(Mixed Integer Programing)***, yang memberikan batasan sebagian variabel adalah *integer*
- ***Pure IP***, jika keseluruhan variabel harus *integer*, dan
- ***Binary IP***, jika seluruh variabel adalah *integer* bernilai 0 atau 1.

Konsep pertama yang perlu diketahui dalam pemecahan masalah *Integer Programming* (dalam hal ini MIP) adalah apa yang disebut dengan relaksasi *(relaxation)*. Relaksasi dari sebuah permasalahan MIP adalah sebuah permasalahan baru sedemikian hingga:

a) Setiap solusi bagi MIP bersesuaian dengan sebuah solusi *feasible* bagi masalah hasil relaksasi.

b) Setiap solusi bagi MIP memiliki nilai fungsi objektif lebih besar atau sama dengan solusi yang bersesuaian pada masalah hasil relaksasi.

Relaksasi yang paling umum digunakan untuk permasalahan MIP adalah relaksasi *Linear Programming (LP relaxation)*. Relaksasi ini identik dengan permasalahan MIP itu sendiri terkecuali bahwa batasan yang mengharuskan solusi berupa *integer* dihilangkan. Oleh karena MIP dan relaksasinya adalah identik, maka setiap solusi integer yang *feasible* pada MIP adalah juga merupakan solusi bagi relaksasi LP-nya dengan nilai *objective function* (fungsi tujuan) yang sesuai pula.

Smith dan Taskin (2008) menganalogikan permasalahan relaksasi ini sebagai kantong yang berisi kelereng oranye dan kelereng biru. Setiap kelereng mewakili solusi bagi *LP relaxation*, namun hanya kelereng oranye yang juga mewakili solusi bagi MIP. Setiap kelereng memiliki berat yang mewakili nilai *objective function*-nya. Jika diasumsikan optimasi yang dilakukan adalah minimasi, maka tugas dari LP adalah mencari kelereng teringan sedangkan tugas IP adalah mencari kelereng oranye teringan.

Konsep selanjutnya adalah algoritma untuk memecahkan permasalahan MIP itu sendiri, yakni *branch-and-bound*. Untuk mendeskripsikannya, Smith dan Taskin (2008) memberikan contoh permasalahan sebagai berikut:

$$Min\ Z = 4x_1 + 6x_2 \tag{2.19}$$

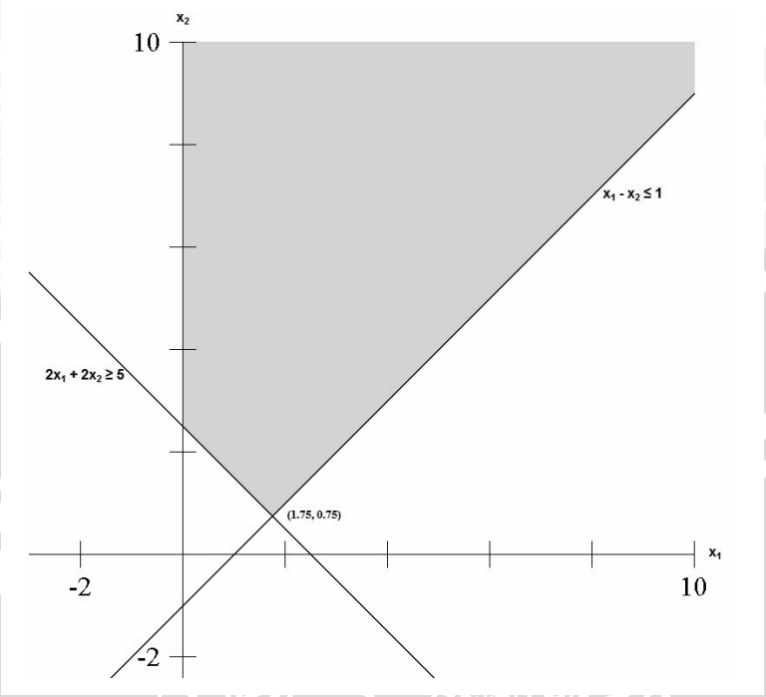
Sedemikian hingga:

$$2x_1 + 2x_2 \geq 5 \tag{2.20a}$$

$$x_1 - x_2 \leq 1 \quad (2.20b)$$

$$x_1, x_2 \geq 0 \text{ dan integer} \quad (2.20c)$$

Gambar 2.3 adalah grafik permasalahan tersebut setelah dilakukan relaksasi:



Gambar 2.3 Grafik permasalahan FIP hasil relaksasi

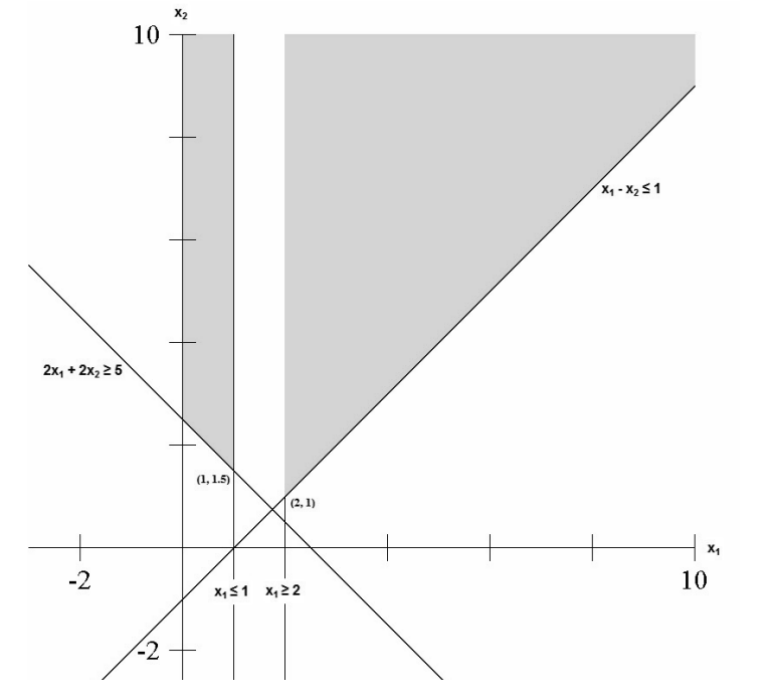
Sumber: Smith dan Taskin (2008)

Dari Gambar 2.3 terlihat bahwa solusi optimal dari *LP relaxation* adalah $(1.75, 0.75)$ dengan nilai *Objective Function* 11.5. Jika kembali pada analogi kelereng biru dan oranye, maka LP telah berhasil menemukan kelereng teringan dalam kantong, yakni seberat 11.5. Artinya, semua kelereng oranye memiliki berat 11.5 atau lebih. Dengan kata lain, batas bawah *(lower bound)* dari solusi optimal MIP adalah 11.5.

Solusi optimal yang dihasilkan *LP relaxation*, yakni $(1.75, 0.75)$, tentu saja tidak *feasible* bagi MIP karena MIP mensyaratkan solusi bernilai integer. Untuk itu, permasalahan ini perlu dibagi menjadi dua sub-permasalahan. Secara grafik, daerah di sekitar solusi *LP relaxation* akan dibagi dua, yakni daerah $x_1 \leq 1$ (selanjutnya disebut *region 1*) dan daerah $x_1 \geq 2$ (selanjutnya disebut *region 2*). Semua solusi MIP terletak tepat di salah satu *region* ini. Hal yang sama dilakukan pula pada x_2 dengan membagi menjadi sub problem $x_2 \leq 0$ dan $x_2 \geq 1$. Proses inilah yang disebut dengan pencabangan *(branching)*.

Secara konsep, *branching* dapat dianalogikan dengan membagi kelereng dalam kantong menjadi dua kantong. Semua kelereng yang mewakili solusi $1 < x < 2$ dibuang, termasuk kelereng teringan yang telah ditemukan sebelumnya. Tidak ada kelereng oranye yang dibuang karena mereka pasti berada di salah satu kantong,

kantong $x_1 \leq 1$ atau kantong $x_1 \geq 2$. Langkah selanjutnya adalah mencari kelereng oranye teringan di masing-masing kantong, membandingkan keduanya, dan mengambil yang teringan di antara keduanya.



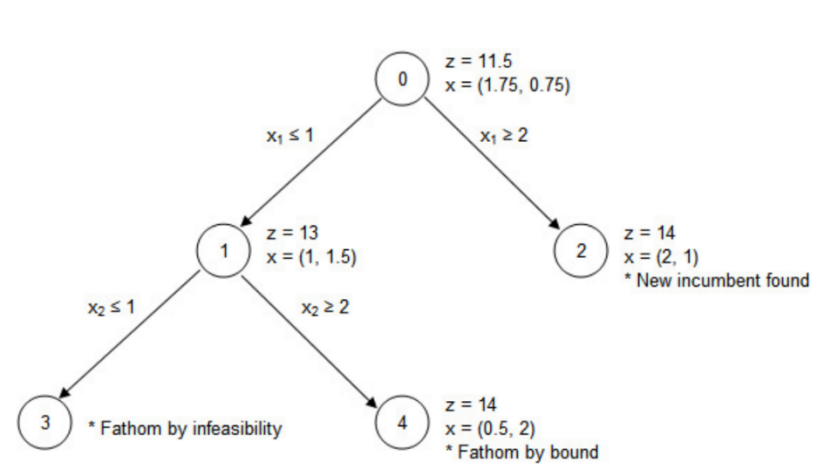
Gambar 2.4 Daerah solusi *feasible* setelah *branching*

Sumber: Smith dan Taskin (2008)

Daerah solusi *feasible* dari kedua sub-permasalahan baru digambarkan pada Gambar 2.4. Untuk daerah $x_1 \leq 1$, solusi optimalnya adalah $(1, 1.5)$ dengan nilai *Objective Function* 13. Sedangkan pada daerah $x_1 \geq 2$, solusi optimalnya adalah $(2, 1)$ dengan nilai *Objective Function* 14. Solusi untuk daerah $x_1 \leq 1$ masih memiliki nilai fraksional (1.5), sehingga nilai *Objective Function* 13 dijadikan sebagai *lower bound*. Di sisi lain, daerah $x_1 \geq 2$ memiliki solusi *integer* untuk kedua variabel sehingga tidak diperlukan pencarian lebih lanjut di daerah tersebut. *region* ini dikatakan sebagi *Fathomed By Integrality*. Solusi (2, 1) disimpan sebagai solusi sementara yang disebut *Incumbent Solution*. Jika tidak ada solusi yang lebih baik dari solusi ini, maka solusi ini akan menjadi solusi optimal.

Pada tahap ini, terdapat satu *region* yang disebut sebagai *region* aktif. *Region* aktif adalah *region* yang belum dilakukan pencabangan dan masih perlu dieksplorasi lebih jauh. Pada contoh ini, *region* awal adalah *region* inaktif, karena sudah dilakukan *branching* pada cabang ini. *Region* 2 juga termasuk inaktif karena solusi integer terbaik untuk daerah tersebut sudah ditemukan *(Fathomed By Integrality)*. Namun, *region* 1 masih aktif dan perlu dieksplorasi lagi. *Lower bound region* ini adalah 13 sehingga dapat diketahui bahwa solusi optimal untuk keseluruhan masalah pasti memiliki nilai *objective function* antara 13 dan 14 (inklusif).

Tahap selanjutnya adalah melakukan *branching* pada *region* yang masih aktif, yakni *region* 1. *Region* ini memiliki solusi optimal $(1, 1.5)$. *Branching* dilakukan dengan membuat dua sub-permasalahan baru, yakni $x_1 \leq 1$ dan $x_2 \leq 1$ (*region* 3) serta $x_1 \leq 1$ dan $x_2 \geq 2$ (*region* 4). Seluruh solusi *integer* yang berada di *region* 1 akan berada tepat di salah satu *region* baru tersebut, *region* 3 atau *region* 4. *Region* 3 tidak memiliki solusi *feasible* dikarenakan syarat $x_1 \leq 1$ dan $x_2 \leq 1$ membuatnya tidak dapat memenuhi batasan ke-2 atau Persamaan 2.20b. Oleh karena itu, tidak ada solusi *integer* di *region* ini dan pencarian di *region* ini dihentikan. *Region* ini dikatakan sebagai *Fathomed By Infeasibility*. *Region* 4 memiliki solusi optimal hasil dari *LP relaxation* $(0.5, 2)$ dengan nilai *Objective Function* 14. Solusi *integer* terbaik pada region ini belum dicari, namun hasil dari *LP relaxation* menunjukkan bahwa *lower bound region* ini adalah 14. Artinya, solusi-solusi di *region* ini memiliki nilai *objective function* sebesar 14 atau lebih. Dikarenakan optimasi yang dilakukan adalah minimasi, dan solusi *incumbent* yang ada saat ini juga menghasilkan *Objective Function* sebesar 14, pencarian di *region* 4 dihentikan. Ini dilakukan karena pencarian di *region* ini tidak akan memberikan perbaikan pada solusi *incumbent*. *Region* ini dikatakan sebagai *Fathomed By Bound*.



Gambar 2.5 Representasi *tree* dari proses *branch-and-bound*

Sumber: Smith dan Taskin (2008)

Gambar 2.5 merupakan representasi dalam bentuk *tree* dari proses ini. Representasi ini disebut juga dengan *branch-and-bound tree*. Setiap *node* pada *tree* mewakili *feasible region*. Kondisi saat ini tidak ada lagi *region* aktif untuk dieksplorasi (tidak ada *node* aktif), algoritma berhenti dan solusi *incumbent* diambil sebagai solusi optimal.

Secara formal, algoritma *branch-and-bound* untuk kasus minimasi dijabarkan sebagai berikut (kasus maksimasi diselesaikan dengan cara yang hampir sama):

- **Langkah 0 (inisiasi)**

  Tetapkan nilai *Objective Function incumbent* $v = \infty$ (kondisi awal di mana tidak ada solusi *integer* yang *feasible*). Tetapkan penghitung *node* aktif *(active node count)* $k = 1$ 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 *LP relaxation* 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 *Objective Function* $z^* \geq v$, 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 bagi *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 *fractional* dari solusi *LP relaxation* 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 *constraint* $x_s \leq \lfloor f \rfloor$ dan yang lain dengan menambahkan *constraint* $x_s \geq \lceil f \rceil$. Tambahkan 1 pada *k* (2 untuk *node* aktif baru minus 1 untuk *branching* pada *current node*) dan kembali ke **Langkah 1**.

Algoritma *branch-and-bound* yang telah diuraikan sebelumnya sebenarnya masih dapat dioptimasi lagi. Misalnya pada **Langkah 0** dapat digunakan prosedur *heuristic* untuk mencari solusi *incumbent* awal secara cepat untuk meningkatkan jumlah *fathomed node* pada **Langkah 4**. Pada **Langkah 2** dapat diterapkan aturan untuk memilih *node* mana yang akan dicabang. Pada **Langkah 6** dapat diterapkan aturan untuk memilih variabel mana yang akan dicabang terlebih dahulu. Langkah-

langkah optimasi tersebut biasanya diterapkan untuk meningkatkan efisiensi algoritma *branch-and-bound* pada *solver* komersial.

2.8 Fuzzy integer programming

Fuzzy Integer Programming (FIP) digunakan untuk memecahkan masalah *Integer Programming* ketika diperlukan *fuzzy number* untuk merepresentasikan data yang tidak presisi. Model *Fuzzy Integer Programming Problem (FIPP)* yang digunakan pada penelitian ini adalah FIPP dengan koefisien fungsi tujuan dan koefisien dari batasan-batasannya adalah *fuzzy number*. Untuk itu digunakan metode pemecahan FIPP sebagai *Multi Objective Integer Programming Problem* (MOIPP) (Dash & Dash, 2012).

Anggap terdapat sebuah FIPP dengan koefisien fungsi tujuan dan koefisien batasan-batasannya berupa *fuzzy number*, maka dapat dituliskan sebagai:

$$Max \sum_{j=1}^{n} \tilde{C}_j x_j \tag{2.21}$$

Sedemikian hingga:

$$\sum_{x \geq 0} (S_{ij}, l_{ij}, r_{ij}) x_{ij} \leq (t_i, u_i, v_i) \tag{2.22}$$

$$0 \leq i \leq m, 0 \leq j \leq n$$

Kembali ke Persamaan 2.18, maka batasan dari FIPP tersebut (Persamaan 2.22) dapat ditulis ulang sebagai:

$$\sum_{j=1}^{n} s_{ij} x_j \leq t_i \tag{2.23a}$$

$$\sum_{j=1}^{n} (s_{ij} - l_{ij}) x_j \leq t_i - u_i \tag{2.23b}$$

$$\sum_{j=1}^{n} (s_{ij} + r_{ij}) x_j \leq t_i + v_i \tag{2.23c}$$

$$x_j \geq 0, \text{ dan integer} \tag{2.23d}$$

Fungsi tujuan dari FIPP (Persamaan 2.21) kemudian dikonversi menjadi MOIPP dengan bentuk:

$$\underset{x \in X}{Max} \{f_1(x), f_2(x), \dots, f_k(x)\} \tag{2.24}$$

Di mana $f_i \ : \ R^n \rightarrow R^i$ dan R^n adalah ruang *Euclidean n*-dimensi. Sehingga FIPP sekarang menjadi:

$$\underset{x \in X}{Max} \sum_{m=1}^{k} f_m(x) \quad (2.25)$$

Sedemikian hingga:

$$\sum_{j=1}^{n} s_{ij} x_j \leq t_i \quad (2.26a)$$

$$\sum_{j=1}^{n} (s_{ij} - l_{ij}) x_j \leq t_i - u_i \quad (2.26b)$$

$$\sum_{j=1}^{n} (s_{ij} + r_{ij}) x_j \leq t_i + v_i \quad (2.26c)$$

$$x_j \geq 0, \text{ dan integer} \quad (2.26d)$$

Berikut ini disajikan contoh pemecahan permasalahan FIP dengan metode konversi ke MOIPP (Dash & Dash, 2012):

$$Max\ Z = (6, 9, 12, 27)x_1 + (15, 20, 25, 30)x_2 \quad (2.27)$$

Dengan batasan:

$$(5, 4, 2)x_1 + (4, 3, 1)x_2 \leq (11, 5, 3) \quad (2.28a)$$

$$(4, 1, 3)x_1 + (5, 4, 1)x_2 \leq (9, 5, 4) \quad (2.28b)$$

Merujuk pada Persamaan 2.26, batasan-batasan FIPP pada Persamaan 2.28 dapat ditulis ulang sebagai:

$$5x_1 + 4x_2 \leq 11$$
$$4x_1 + 5x_2 \leq 9$$
$$x_1 + x_2 \leq 6$$
$$3x_1 + x_2 \leq 4$$
$$7x_1 + 5x_2 \leq 14$$
$$7x_1 + 6x_2 \leq 13$$
$$x_1, x_2 \geq 0, \text{ dan integer} \quad (2.29)$$

Fungsi tujuan dari FIPP (Persamaan 2.27) dapat ditulis dalam bentuk MOIPP sebagai:

$$Max\ Z = (6x_1 + 15x_2) + (9x_1 + 20x_2) + (12x_1 + 25x_2) + (27x_1 + 30x_2)$$
$$= 54x_1 + 90x_2 \quad (2.30)$$

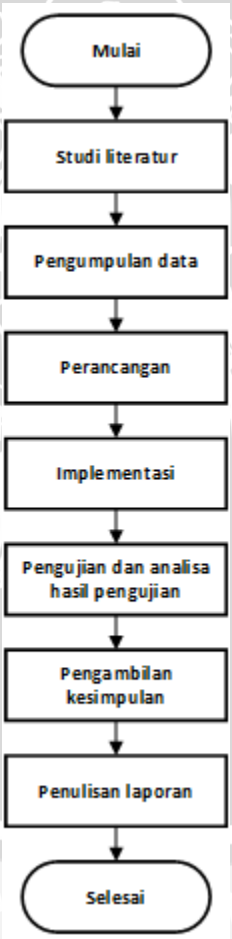
Setelah proses konversi ini, maka permasalahan menjadi permasalahan *Integer Programming* biasa, dengan fungsi tujuan Persamaan 2.30 dan batasan-batasan Persamaan 2.29. Setelah diselesaikan dengan metode *Integer Programming*, didapatkan solusi $x_1 = 1$ dan $x_2 = 1$ dengan fungsi tujuan berupa *fuzzy number* $(21, 29, 37, 57)$.

BAB 3 METODOLOGI

Bab ini berisi pembahasan mengenai metodologi penelitian yang akan dilakukan dalam usaha untuk menerapkan dan menguji algoritma *Fuzzy Integer Programming* untuk keperluan optimasi kandungan gizi menu harian. Pembahasan dalam bab ini meliputi tipe dan strategi penelitian, contoh perhitungan manual, teknik pengumpulan data, serta pembahasan mengenai strategi pengujian dan analisis hasil pengujian.

3.1 Tipe dan strategi penelitian

Penelitian ini adalah penelitian bertipe implementatif pengembangan di mana penulis ingin menerapkan dan menguji algoritma *Fuzzy Integer Programming* untuk keperluan optimasi menu harian. Untuk itu diperlukan tahapan-tahapan sistematis agar dapat menghasilkan sebuah penelitian yang akurat dan dapat dipertanggungjawabkan. Tahapan-tahapan tersebut secara umum dapat digambarkan dalam bentuk diagram pada Gambar 3.1.



Gambar 3.1 Diagram tahapan-tahapan penelitian

- **Studi literatur**

  Tahap ini diperlukan untuk mendapatkan gambaran tentang apa yang pernah dikerjakan peneliti lain sehubungan dengan topik penelitian serta bagaimana mereka mengerjakannya. Hal ini penting untuk membandingkan antara apa yang akan dikerjakan dengan apa yang telah dikerjakan peneliti-peneliti lain serta untuk menghindari usaha yang sebelumnya pernah dicoba. Pada tahap studi literatur dilakukan pula studi mengenai teori-teori dasar, konsep, serta metode-metode yang berkaitan dengan penelitian ini. Konsep-konsep dasar ini diperlukan sebagai landasan dari metode yang sedang dikembangkan serta mempermudah pembaca dalam memahami permasalahan yang diangkat penulis.

- **Pengumpulan data**

  Pada tahap ini, data-data yang diperlukan untuk menerapkan dan menguji sistem akan dikumpulkan. Strategi pengumpulan data yang akan digunakan akan dibahas pada bagian "Pengumpulan Data".

- **Perancangan**

  Pada tahap ini sistem akan mulai dirancang. Perancangan diperlukan agar pengembangan sistem dapat lebih terorganisir dan terarah. Tahap perancangan terdiri atas perancangan arsitektur sistem secara umum, perancangan kelas-kelas utama, perancangan basis data, serta perancangan antarmuka pengguna *(user interface)*.

- **Implementasi**

  Tahap selanjutnya setelah perancangan sistem adalah mengimplementasikan rancangan tersebut ke dalam bentuk sebuah aplikasi. Aplikasi hasil implementasi juga harus dapat mendukung proses pengujian algoritma.

- **Pengujian dan analisa hasil pengujian**

  Pada tahap ini, aplikasi yang telah selesai akan digunakan sebagai *tool* untuk menguji algoritma yang diusulkan. Pengujian akan dilakukan dengan data uji dan berdasarkan strategi pengujian yang telah dirancang. Tujuannya adalah untuk dapat menjawab pertanyaan-pertanyaan yang tertuang dalam rumusan masalah penelitian.

- **Pengambilan kesimpulan**

  Setelah mendapatkan data hasil pengujian serta analisanya, maka satu atau beberapa kesimpulan dapat ditarik dari hasil tersebut. Tujuannya adalah memberikan poin-poin penting berkenaan dengan hasil yang didapat dari keseluruhan proses penelitian. Untuk menambah kemanfaatan penelitian, saran mengenai arah pengembangan selanjutnya juga disertakan.

- **Penulisan laporan**

  Laporan penelitian ditulis setelah semua tahapan penelitian telah dilalui. Tujuan pembuatan laporan adalah untuk mendokumentasikan seluruh proses penelitian. Laporan penelitian diharapkan dapat menjadi bahan rujukan bagi pengembangan sistem, algoritma atau metode-metode selanjutnya.

3.2 Contoh perhitungan manual

Pada bagian ini akan disajikan contoh perhitungan manual yang menunjukkan bagaimana proses optimasi menu harian dengan FIP bekerja. Data yang digunakan dalam perhitungan ini tidak berdasarkan data sebenarnya dan semata-mata hanya untuk mempermudah ilustrasi.

Masukkan utama bagi sistem adalah sebuah *menu set*. *Menu set* ini berisi *id* dari menu-menu yang dipilih oleh pengguna. Sistem akan melakukan *query* ke basis data untuk meminta data dari tiap menu *id* yang ada dalam *menu set* ini. Anggap *menu set* yang akan dioptimasi berisi *id* dari menu "Soto Ayam", "Nasi" dan "Jus Alpokat". Setelah melakukan *query*, sistem akan menerima data dari basis data yang dapat digambarkan dalam bentuk tabel pada Tabel 3.1, Tabel 3.2 dan Tabel 3.3.

Tabel 3.1 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			
Bahan utama	***Jumlah***	***Faktor konversi***	***Berat hasil konversi***	***Berat fuzzy (s, l,r)***
- 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)
Bahan tambahan	***Jumlah***	***Faktor konversi***	***Berat hasil konversi***	***Berat fuzzy (s, l,r)***
- Lengkuas	1 ruas jari	-	-	-
- Daun jeruk	3 lembar	-	-	-

Tabel 3.2 Data menu nasi

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

Tabel 3.3 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			
Bahan utama	***Jumlah***	***Faktor konversi***	***Berat hasil konversi***	***Berat fuzzy (s, l,r)***
- 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)
Bahan tambahan	***Jumlah***	***Faktor konversi***	***Berat hasil konversi***	***Berat fuzzy (s, l,r)***
- Air Matang	250 ml	-	-	-
- Es Batu	150 gr	-	-	-

3.2.1 Perhitungan nutrisi menu

Hal pertama yang perlu dilakukan pada data menu yang telah diterima dari basis data adalah menghitung kandungan nutrisinya. Pada tahap ini, terlebih dahulu dilakukan normalisasi tiap menu terpilih menjadi 1 porsi. Hal ini karena beberapa resep menu adalah resep untuk lebih dari 1 porsi. Pada contoh ini, resep Soto Ayam (Tabel 3.1) adalah untuk 4 porsi, maka dengan melakukan perkalian skalar pada nilai *fuzzy* masing-masing berat bahan didapatkan:

- **Soto Ayam**

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

  Kol : ¼ $\times (200, 50, 50) = (50, 12.5, 12.5)$ gram

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

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

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

Selanjutnya, dilakukan perhitungan kandungan gizi untuk masing-masing bahan dari tiap menu dan menjumlahkannya menjadi kandungan gizi menu. Untuk mempermudah contoh perhitungan, maka nutrien yang diikutsertakan adalah energi, protein, lemak, vitamin A, vitamin B, dan vitamin C. Data kandungan gizi bahan dari basis data adalah per 100 gram, sehingga perlu dikonversi menjadi per 1 gram terlebih dahulu sebelum dikalikan dengan berat *fuzzy* bahan. Misalkan dari basis data didapatkan kandungan energi ayam per 100 gram adalah 302 *Kcal*. Berat *fuzzy* bahan ayam dari menu Soto Ayam per porsi adalah $(37.5, 2.5, 5)$ gram, sehingga untuk mencari nilai energi per porsi bahan ayam dari menu Soto Ayam perhitungannya adalah sebagai berikut:

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

Dengan cara yang sama, didapatkan kandungan gizi tiap menu adalah sebagai berikut:

Tabel 3.4 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (1)

Bahan	Berat *fuzzy*			Data dari DB					
	s	l	r	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Ayam	37.5	2.5	5	302	18.2	25	810	0.08	0
- Kol	50	12.5	12.5	22	2.1	0.5	0	0.03	3
- Bihun	12.5	1.25	1.25	360	4.7	0.1	0	0	0
- Telur	22.5	12.5	2.5	174	10.8	14	0	0.78	0
- Tomat	50	1.25	2.5	20	1	0.3	1500	0.06	40

Tabel 3.5 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (2)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Energi	Protein	Lemak
- Ayam	(113.25, 7.55 ,15.1)	(6.825, 0.455 ,0.91)	(9.375, 0.625 ,1.25)
- Kol	(11, 2.75 ,2.75)	(1.05, 0.2625 ,0.2625)	(0.25, 0.0625 ,0.0625)
- Bihun	(45, 4.5 ,4.5)	(0.5875, 0.05875 ,0.05875)	(0.0125, 0.00125 ,0.00125)
- Telur	(39.15, 21.75 ,4.35)	(2.43, 1.35 ,0.27)	(3.15, 1.75 ,0.35)
- Tomat	(10, 0.25 ,0.5)	(0.5, 0.0125 ,0.025)	(0.15, 0.00375 ,0.0075)
Total	*(218.4, 36.8, 27.2)*	*(11.39, 2.14, 1.53)*	*(12.94, 2.44, 1.67)*

Tabel 3.6 Perhitungan nilai *fuzzy* kandungan gizi soto ayam (3)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Vit. A	Vit. B	Vit. C
- Ayam	(303.75, 20.25 ,40.5)	(0.03, 0.002 ,0.004)	(0, 0 ,0)
- Kol	(0, 0 ,0)	(0.015, 0.00375 ,0.00375)	(1.5, 0.375 ,0.375)
- Bihun	(0, 0 ,0)	(0, 0 ,0)	(0, 0 ,0)
- Telur	(0, 0 ,0)	(0.1755, 0.0975 ,0.0195)	(0, 0 ,0)
- Tomat	(750, 18.75 ,37.5)	(0.03, 0.00075 ,0.0015)	(20, 0.5 ,1)
Total	*(1053.75, 39, 78)*	*(0.25, 0.1, 0.03)*	*(21.5, 0.88, 1.38)*

Tabel 3.7 Perhitungan nilai *fuzzy* kandungan gizi nasi (1)

Bahan	Berat Fuzzy			Data dari DB					
	s	l	r	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Nasi	200	50	100	176	3.6	0.4	0	0.1	0

Tabel 3.8 Perhitungan nilai *fuzzy* kandungan gizi nasi (2)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Energi	Protein	Lemak
- Nasi	(352, 88 ,176)	(7.2, 1.8 ,3.6)	(0.8, 0.2 ,0.4)
Total	*(352, 88 ,176)*	*(7.2, 1.8 ,3.6)*	*(0.8, 0.2 ,0.4)*

Tabel 3.9 Perhitungan nilai *fuzzy* kandungan gizi nasi (3)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Vit. A	Vit. B	Vit. C
- Nasi	(0, 0 ,0)	(0.2, 0.05 ,0.1)	(0, 0 ,0)
Total	*(0, 0 ,0)*	*(0.2, 0.05 ,0.1)*	*(0, 0 ,0)*

Tabel 3.10 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (1)

Bahan	Berat Fuzzy			Data dari DB					
	s	l	r	Energi	Protein	Lemak	Vit. A	Vit. B	Vit. C
- Alpokat	1200	100	200	85	0.9	6.5	180	0.05	13
- Gula Pasir	100	50	50	364	0	0	0	0	0
- Susu Kental Mns.	100	20	50	336	8.2	10	510	0.05	1

Tabel 3.11 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (2)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Energi	Protein	Lemak
- Alpokat	(1020, 85 ,170)	(10.8, 0.9 ,1.8)	(78, 6.5 ,13)
- Gula Pasir	(364, 182 ,182)	(0, 0 ,0)	(0, 0 ,0)
- Susu Kental Mns.	(336, 67.2 ,168)	(8.2, 1.64 ,4.1)	(10, 2 ,5)
Total	*(1720, 334.2, 520)*	*(19, 2.54, 5.9)*	*(88, 8.5, 18)*

Tabel 3.12 Perhitungan nilai *fuzzy* kandungan gizi jus alpokat (3)

Bahan	Kandungan Gizi Fuzzy (s, l, r)		
	Vit. A	Vit. B	Vit. C
- Alpokat	(2160, 180 ,360)	(0.6, 0.05 ,0.1)	(156, 13 ,26)
- Gula Pasir	(0, 0 ,0)	(0, 0 ,0)	(0, 0 ,0)
- Susu Kental Mns.	(510, 102 ,255)	(0.05, 0.01 ,0.025)	(1, 0.2 ,0.5)
Total	*(2670, 282, 615)*	*(0.65, 0.06, 0.125)*	*(157, 13.2, 26.5)*

3.2.2 Pembentukan persamaan FIPP

Tahap pertama pembentukan persamaan *Fuzzy Integer Programming Problem* (FIPP) adalah pembentukan persamaan batasan atau *constraint*. FIP hanya dapat menentukan apakah suatu menu disajikan atau tidak. Untuk menentukan seberapa banyak menu akan disajikan, angka-angka dalam data porsi penyajian normal akan dimanfaatkan. Berdasarkan data ini, satu menu akan digandakan dengan ukuran berbeda-beda. Tugas FIP selanjutnya tinggal memilih salah satu dari hasil penggandaan ini untuk dapat menghasilkan nilai fungsi tujuan yang optimal. Misalkan terdapat Menu $M1$ dan $M2$ dengan porsi penyajian normal masing-masing adalah $[1, ½, ¼]$ porsi dan $[1, ¼]$ porsi, maka persamaan batasan bawahnya adalah:

$$a_{11}M11 + a_{12}M12 + a_{13}M13 + a_{14}M21 + a_{15}M22 \geq b_1 \quad (3.1a)$$

$$a_{21}M11 + a_{22}M12 + a_{23}M13 + a_{24}M21 + a_{25}M22 \geq b_2 \quad (3.1b)$$

Di mana $M11$ = 1 porsi $M1$, $M12$ = ½ porsi $M1$, $M13$ = ¼ porsi $M1$, $M21$ = 1 porsi $M2$ dan $M22$ = ¼ porsi $M2$. $b1$ dan $b2$ adalah angka batasan nutrien atau dapat dikatakan kebutuhan gizi subjek.

Nantinya, pada aplikasi hasil implementasi akan dibentuk pula batasan-batasan atas untuk membatasi nutrien agar tidak melebihi nilai UL *(Upper Level)*. Persamaan batasan-batasan ini mirip dengan persamaan batasan untuk batas-batas bawah (Persamaan 3.1), hanya saja dengan tanda pertidaksamaan $\leq$. Namun, untuk menyederhanakan perhitungan, batasan-batasan ini tidak diikutsertakan dalam contoh perhitungan.

Selanjutnya, perlu dihitung nilai EER dan RDA. Anggap subjek adalah laki-laki dengan usia (U) 20 tahun, tinggi badan (TB) 163 cm, berat badan (BB) 50 kg dan aktivitas fisik "Aktif". Dari Tabel 2.3 diperoleh koefisien PA subjek adalah 1.25. Berdasarkan rumus perhitungan EER pada Tabel 2.2 nilai EER subjek adalah:

$$[662 - (9.53 \times U)] + PA \times [(15.91 \times BB) + (539.6 \times TB)] = 2565.21\ Kcal$$

Untuk mendapatkan data RDA subyek, sistem akan melakukan *query* ke basis data. Data dari basis data adalah nilai RDA untuk setiap nutrien yang diikutsertakan. Berdasarkan data RDA didapatkan nilai kecukupan gizi subyek adalah sebagai berikut:

- Protein = 56 g/d
- Lemak = Tidak ada data
- Vitamin A = 900 µg/d
- Vitamin B = 1.2 mg/d
- Vitamin C = 90 mg/d

Namun demikian, untuk contoh permasalahan yang akan diselesaikan ini, angka-angka batasan tersebut masih terlalu tinggi mengingat menu yang dipilih sebagai contoh dapat dikatakan "tidak sehat". Untuk itu, angka batasan akan diturunkan sekitar setengah dari nilai aslinya. Hal ini semata-mata hanya untuk mempermudah proses ilustrasi. Berikut adalah nilai batasan-batasan gizi yang akan digunakan sebagai contoh:

- Energi = 1000 kcal
- Protein = 25 g/d
- Vitamin A = 450 µg/d
- Vitamin B = 0.5 mg/d
- Vitamin C = 45 mg/d

Sebelum diubah menjadi persamaan batasan, data RDA ini akan terlebih dahulu diubah menjadi sebuah *fuzzy number* untuk membuat batasan lebih fleksibel. Tingkat 'fleksibilitas' batasan ini diatur oleh parameter yang disebut sebagai 'keburaman batasan gizi' atau *'reference fuzziness'*. Parameter ini menentukan berapa persen nilai *left spread* (l) dan/atau *right spread* (r) dari nilai pastinya (s). Hal ini akan memberikan sedikit toleransi pada batasan-batasan yang ada. Seperti dibahas pada bagian dasar teori, kelebihan atau kekurangan sedikit di atas atau di bawah RDA/AI tidak menimbulkan dampak berarti. Pemberian toleransi ini juga untuk mengurangi kemungkinan *solver* berakhir pada status *infeasible*. Nantinya, pengaruh pemberian toleransi ini juga akan diuji pada bagian pengujian untuk mengetahui pengaruhnya terhadap hasil optimasi. Pada contoh ini, parameter *'reference fuzziness'* akan diberi nilai 15%. Maka, untuk batasan energi dengan angka RDA 1000 *Kcal* akan menghasilkan sebuah *fuzzy number*:

- *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 batasan sebagai berikut:

- **Energi**

$$2 \times (218.4,\ 36.8,\ 27.2)M11 + 1 \times (218.4,\ 36.8,\ 27.2)M12 + \frac{1}{2} \times (218.4,\ 36.8,\ 27.2)M13 + \frac{1}{4} \times (218.4,\ 36.8,\ 27.2)M14 + 1 \times (352, 88, 176)M21 + 1 \times (1720, 334.2, 520)M31 + \frac{1}{2} \times (1720, 334.2, 520)M32 \geq (1000, 150, 150) \quad (3.2a)$$

Atau

$$(436.8,\ 73.6,\ 54.4)M11 + (218.4,\ 36.8,\ 27.2)M12 + (109.2,\ 18.4,\ 13.6)M13 + (54.6,\ 9.2,\ 6.8)M14 + (352, 88, 176)M21 + (1720, 334.2, 520)M31 + (860, 167.1, 260)M32 \geq (1000, 150, 150) \quad (3.2b)$$

- **Protein**

$2 \times (11.39,\ 2.14,\ 1.53)M11 + 1 \times (11.39,\ 2.14,\ 1.53)M12 + \frac{1}{2} \times (11.39,\ 2.14,\ 1.53)M13 + \frac{1}{4} \times (11.39,\ 2.14,\ 1.53)M14 + 1 \times (7.2,\ 1.8, 3.6)M21 + 1 \times (19, 2.54, 5.9)M31 + \frac{1}{2} \times (19, 2.54, 5.9)M32 \geq (25, 3.75, 3.75)$ (3.3a)

Atau

$(22.78,\ 4.28,\ 3.06)M11 + (11.39,\ 2.14,\ 1.53)M12 + (5.69,\ 1.07,\ 0.76)M13 + (2.85,\ 0.54,\ 0.38)M14 + (7.2,\ 1.8, 3.6)M21 + (19, 2.54, 5.9)M31 + (9.5, 1.27, 2.95)M32 \geq (25, 3.75, 3.75)$ (3.3b)

- **Lemak**

Karena acuan RDA/AI tidak memiliki data batasan untuk lemak, maka tidak perlu membentuk batasan untuk nutrien ini.

- **Vitamin A**

$2 \times (1053.75,\ 39,\ 78)M11 + 1 \times (1053.75,\ 39,\ 78)M12 + \frac{1}{2} \times (1053.75,\ 39,\ 78)M13 + \frac{1}{4} \times (1053.75,\ 39,\ 78)M14 + 1 \times (0, 0, 0)M21 + 1 \times (2670, 282, 615)M31 + \frac{1}{2} \times (2670, 282, 615)M32 \geq (450, 67.5, 67.5)$ (3.4a)

Atau

$(2107.5,\ 78,\ 156)M11 + (1053.75,\ 39,\ 78)M12 + (526.87,\ 19.5,\ 39)M13 + (263.44,\ 9.75,\ 19.5)M14 + (0, 0, 0)M21 + (2670, 282, 615)M31 + (1335, 141, 307.5)M32 \geq (450, 67.5, 67.5)$ (3.4b)

- **Vitamin B**

$2 \times (0.25,\ 0.1,\ 0.03)M11 + 1 \times (0.25,\ 0.1,\ 0.03)M12 + \frac{1}{2} \times (0.25,\ 0.1,\ 0.03)M13 + \frac{1}{4} \times (0.25,\ 0.1,\ 0.03)M14 + 1 \times (0.2, 0.05, 0.1)M21 + 1 \times (0.65, 0.06, 0.13)M31 + \frac{1}{2} \times (0.65, 0.06, 0.13)M32 \geq (0.5, 0.075, 0.075)$ (3.5a)

Atau

$$(0.5,\ 0.2,\ 0.06)M11 + (0.25,\ 0.1,\ 0.03)M12 + (0.125,\ 0.05,\ 0.015)M13 + (0.062,\ 0.025,\ 0.0075)M14 + (0.2, 0.05, 0.1)M21 + (0.65, 0.06, 0.13)M31 + (0.33, 0.03, 0.06)M32 \geq (0.5, 0.075, 0.075) \quad (3.5b)$$

- **Vitamin C**

$$2 \times (21.5,\ 0.88,\ 1.38)M11 + 1 \times (21.5,\ 0.88,\ 1.38)M12 + \frac{1}{2} \times (21.5,\ 0.88,\ 1.38)M13 + \frac{1}{4} \times (21.5,\ 0.88,\ 1.38)M14 + 1 \times (0, 0, 0)M21 + 1 \times (157, 13.2, 26.5)M31 + \frac{1}{2} \times (157, 13.2, 26.5)M32 \geq (45, 6.75, 6.75) \quad (3.6a)$$

Atau

$$(43,\ 1.76,\ 2.76)M11 + (21.5,\ 0.88,\ 1.38)M12 + (10.75,\ 0.44,\ 0.69)M13 + (5.37,\ 0.22,\ 0.35)M14 + (0, 0, 0)M21 + (157, 13.2, 26.5)M31 + (78.5, 6.6, 13.25)M32 \geq (45, 6.75, 6.75) \quad (3.6b)$$

Selain batasan-batasan tersebut, diperlukan lagi beberapa batasan tambahan. Batasan tambahan pertama adalah batasan untuk menjaga total harga agar tidak melebihi anggaran. Batasan ini dibentuk berdasarkan harga *fuzzy* tiap kombinasi menu dan porsi penyajian normal dari menu tersebut. Jika dianggap anggaran yang ditetapkan adalah sebesar Rp. 30.000, maka batasan tersebut dapat dituliskan dalam bentuk persamaan sebagai berikut:

- **Anggaran**

$$(16000, 1000, 1000)M11 + (8000, 500, 500)M12 + (4000, 250, 250)M13 + (2000, 125, 125)M14 + (4000, 500, 500)M21 + (6000, 500, 500)M31 + (3000, 250, 250)M32 \leq (30000, 0, 0) \quad (3.7)$$

Beberapa batasan juga diperlukan untuk memastikan setiap menu dalam menu set yang dipilih disajikan tepat 1 kali. Batasan-batasan tersebut adalah sebagai berikut:

- **Untuk Menu 1 (Soto Ayam)**

$$M11 + M12 + M13 + M14 = 1 \quad (3.8)$$

- **Untuk Menu 2 (Nasi)**

$$M21 = 1 \quad (3.9)$$

- **Untuk Menu 3 (Jus Alpokat)**

$$M31 + M32 = 1 \quad (3.10)$$

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*. Batasan tersebut dapat dituliskan sebagai berikut:

- ***Integrality constraints***

$$M11, M12, M13, M14, M21, M31, M32 \geq 0, integer \quad (3.11)$$

Setelah semua persamaan batasan tersebut terbentuk, tahap terakhir pembentukan persamaan FIPP adalah membentuk persamaan fungsi tujuan. Persamaan fungsi tujuan dibentuk berdasarkan data harga *fuzzy* tiap kombinasi menu dan porsi penyajian normal dari menu tersebut. Tujuan yang ingin dicapai adalah mendapatkan kombinasi dengan harga paling murah, sehingga optimasi yang dilakukan adalah minimasi. Dengan demikian, akan terbentuk persamaan fungsi tujuan sebagai berikut:

- **Fungsi Tujuan**

$$Min.Z = 2 \times (8000, 500, 500)M11 + 1 \times (8000, 500, 500)M12 + \frac{1}{2} \times (8000, 500, 500)M13 + \frac{1}{4} \times (8000, 500, 500)M14 + 1 \times (4000, 500, 500)M21 + 1 \times (6000, 500, 500)M31 + \frac{1}{2} \times (6000, 500, 500)M32 \quad (3.12a)$$

Atau

$$Min.Z = (16000, 1000, 1000)M11 + (8000, 500, 500)M12 + (4000, 250, 250)M13 + (2000, 125, 125)M14 + (4000, 500, 500)M21 + (6000, 500, 500)M31 + (3000, 250, 250)M32 \quad (3.12b)$$

3.2.3 Konversi FIPP ke MOIPP

FIPP yang telah terbentuk pada tahap pembentukan persamaan FIPP selanjutnya akan dikonversi ke bentuk MOIPP agar dapat diselesaikan dengan menggunakan metode *Integer Programming* biasa. Berdasarkan Persamaan 2.25, fungsi tujuan pada Persamaan 3.12b dapat dikonversi menjadi:

$$Min.Z = 18000M11 + 9000M12 + 4500M13 + 2250M14 + 5000M21 + 7000M31 + 3500M32 \quad (3.13)$$

Konversi juga dilakukan pada batasan-batasan FIPP. Berdasarkan Persamaan 2.26, sebuah persamaan batasan yang melibatkan *fuzzy number* dalam FIPP akan dikonversi menjadi 3 buah persamaan batasan. Berikut adalah hasil konversi dari batasan-batasan tersebut:

- **Persamaan 3.2b dikonversi menjadi**

$$436.8M11 + 218.4M12 + 109.2M13 + 54.6M14 + 352M21 + 1720M31 + 860M32 \geq 1000 \quad (3.14)$$

$$363.2M11 + 181.6M12 + 90.8M13 + 45.4M14 + 264M21 + 1385.8M31 + 692.9M32 \geq 850 \quad (3.15)$$

$$491.2M11 + 245.6M12 + 122.8M13 + 61.4M14 + 528M21 + 2240M31 + 1120M32 \geq 1150 \quad (3.16)$$

- **Persamaan 3.3b dikonversi menjadi**

$$22.78M11 + 11.39M12 + 5.69M13 + 2.85M14 + 7.2M21 + 19M31 + 9.5M32 \geq 25 \quad (3.17)$$

$$18.5M11 + 9.25M12 + 4.62M13 + 2.31M14 + 5.4M21 + 16.46M31 + 8.23M32 \geq 21.25 \quad (3.18)$$

$$25.84M11 + 12.92M12 + 6.45M13 + 3.23M14 + 10.8M21 + 24.9M31 + 12.45M32 \geq 28.75 \quad (3.19)$$

- **Persamaan 3.4b dikonversi menjadi**

$$2107.5M11 + 1053.75M12 + 526.87M13 + 263.44M14 + 0M21 + 2670M31 + 1335M32 \geq 450 \quad (3.20)$$

$$2029.5M11 + 1014.75M12 + 507.37M13 + 253.69M14 + 0M21 + 2388M31 + 1194M32 \geq 382.5 \quad (3.21)$$

$$2263.5M11 + 1131.75M12 + 565.87M13 + 282.94M14 + 0M21 + 3285M31 + 1642.5M32 \geq 517.5 \quad (3.22)$$

- **Persamaan 3.5b dikonversi menjadi**

$$0.5M11 + 0.25M12 + 0.13M13 + 0.06M14 + 0.2M21 + 0.65M31 + 0.33M32 \geq 0.5 \quad (3.23)$$

$$0.3M11 + 0.15M12 + 0.08M13 + 0.04M14 + 0.15M21 + 0.59M31 + 0.3M32 \geq 0.425 \quad (3.24)$$

$$0.56M11 + 0.28M12 + 0.14M13 + 0.07M14 + 0.3M21 + 0.78M31 + 0.39M32 \geq 0.575 \quad (3.25)$$

- **Persamaan 3.6b dikonversi menjadi**

$$43M11 + 21.5M12 + 10.75M13 + 5.37M14 + 0M21 + 157M31 + 78.5M32 \geq 45 \quad (3.26)$$

$$41.24M11 + 20.62M12 + 10.31M13 + 5.15M14 + 0M21 + 143.8M31 + 71.9M32 \geq 38.25 \quad (3.27)$$

$$45.76M11 + 22.88M12 + 11.44M13 + 5.72M14 + 0M21 + 183.5M31 + 91.75M32 \geq 51.75 \quad (3.28)$$

- **Persamaan 3.7 dikonversi menjadi**

$$16000M11 + 8000M12 + 4000M13 + 2000M14 + 4000M21 + 6000M31 + 3000M32 \leq 30000 \quad (3.29)$$

$$15000M11 + 7500M12 + 3750M13 + 1875M14 + 3500M21 + 5500M31 + 2750M32 \leq 30000 \quad (3.30)$$

$$17000M11 + 8500M12 + 4250M13 + 2125M14 + 4500M21 + 6500M31 + 3250M32 \leq 30000 \quad (3.31)$$

Batasan-batasan lain yang tidak melibatkan *fuzzy number* tidak dikonversi sehingga tetap dalam bentuk sebagai berikut:

$$M11 + M12 + M13 + M14 = 1 \quad (3.32)$$

$$M21 = 1 \quad (3.33)$$

$$M31 + M32 = 1 \quad (3.34)$$

$$M11, M12, M13, M14, M21, M31, M32 \geq 0, integer \quad (3.35)$$

Fungsi tujuan pada Persamaan 3.13 dan batasan-batasan pada Persamaan 3.14 hingga Persamaan 3.35 inilah yang akan berusaha dipecahkan oleh *Integer Programming*.

3.2.4 Pemecahan dengan *integer programming*

Setelah fungsi tujuan dan batasan-batasan dikonversi, maka permasalahan tersebut dapat dipecahkan dengan metode *Integer Programming* biasa. Langkah pertama adalah melakukan relaksasi pada permasalahan dengan cara menghilangkan batasan-batasan yang mengharuskan solusi bernilai *integer*. Hasil dari relaksasi adalah model sebagai berikut:

$$Min.Z = 18000M11 + 9000M12 + 4500M13 + 2250M14 + 5000M21 + 7000M31 + 3500M32 \quad (3.36)$$

Dengan batasan:

$$436.8M11 + 218.4M12 + 109.2M13 + 54.6M14 + 352M21 + 1720M31 + 860M32 \geq 1000 \quad (3.37)$$

$$363.2M11 + 181.6M12 + 90.8M13 + 45.4M14 + 264M21 + 1385.8M31 + 692.9M32 \geq 850 \quad (3.38)$$

$$491.2M11 + 245.6M12 + 122.8M13 + 61.4M14 + 528M21 + 2240M31 + 1120M32 \geq 1150 \quad (3.39)$$

$$22.78M11 + 11.39M12 + 5.69M13 + 2.85M14 + 7.2M21 + 19M31 + 9.5M32 \geq 25 \quad (3.40)$$

$$18.5M11 + 9.25M12 + 4.62M13 + 2.31M14 + 5.4M21 + 16.46M31 + 8.23M32 \geq 21.25 \quad (3.41)$$

$$25.84M11 + 12.92M12 + 6.45M13 + 3.23M14 + 10.8M21 + 24.9M31 + 12.45M32 \geq 28.75 \quad (3.42)$$

$$2107.5M11 + 1053.75M12 + 526.87M13 + 263.44M14 + 0M21 + 2670M31 + 1335M32 \geq 450 \quad (3.43)$$

$$2029.5M11 + 1014.75M12 + 507.37M13 + 253.69M14 + 0M21 + 2388M31 + 1194M32 \geq 382.5 \quad (3.44)$$

$$2263.5M11 + 1131.75M12 + 565.87M13 + 282.94M14 + 0M21 + 3285M31 + 1642.5M32 \geq 517.5 \quad (3.45)$$

$$0.5M11 + 0.25M12 + 0.13M13 + 0.06M14 + 0.2M21 + 0.65M31 + 0.33M32 \geq 0.5 \quad (3.46)$$

$$0.3M11 + 0.15M12 + 0.08M13 + 0.04M14 + 0.15M21 + 0.59M31 + 0.3M32 \geq 0.425 \quad (3.47)$$

$$0.56M11 + 0.28M12 + 0.14M13 + 0.07M14 + 0.3M21 + 0.78M31 + 0.39M32 \geq 0.575 \quad (3.48)$$

$$43M11 + 21.5M12 + 10.75M13 + 5.37M14 + 0M21 + 157M31 + 78.5M32 \geq 45 \quad (3.49)$$

$$41.24M11 + 20.62M12 + 10.31M13 + 5.15M14 + 0M21 + 143.8M31 + 71.9M32 \geq 38.25 \quad (3.50)$$

$$45.76M11 + 22.88M12 + 11.44M13 + 5.72M14 + 0M21 + 183.5M31 + 91.75M32 \geq 51.75 \quad (3.51)$$

$$16000M11 + 8000M12 + 4000M13 + 2000M14 + 4000M21 + 6000M31 + 3000M32 \leq 30000 \quad (3.52)$$

$$15000M11 + 7500M12 + 3750M13 + 1875M14 + 3500M21 + 5500M31 + 2750M32 \leq 30000 \quad (3.53)$$

$$17000M11 + 8500M12 + 4250M13 + 2125M14 + 4500M21 + 6500M31 + 3250M32 \leq 30000 \quad (3.54)$$

$$M11 + M12 + M13 + M14 = 1 \quad (3.55)$$

$$M21 = 1 \quad (3.56)$$

$$M31 + M32 = 1 \quad (3.57)$$

$$M11, M12, M13, M14, M21, M31, M32 \geq 0 \quad (3.58)$$

Tahap selanjutnya adalah tahap inisiasi. Pada tahap ini, dibuat sebuah *initial node* yang berisi permasalahan awal. *Node* ini dinamai sebagai *region* 1. Nilai *incumbent solution* saat ini adalah *infinity* karena belum menemukan solusi sama sekali. Setelah tahap inisiasi selesai, *solver* mulai melakukan iterasi untuk pemecahan masalah.

3.2.4.1 Iterasi-1

Iterasi diawali dengan pencarian *node* aktif. Saat ini, satu-satunya *node* aktif adalah *initial node* sehingga dilakukan pemecahan dengan *Linear Programming (LP Solving)* pada *node* ini. Dengan bantuan perangkat lunak bernama "Lingo" (Lindo Systems Inc., 2016) diketahui bahwa solusi optimal untuk permasalahan LP pada *region* 1 adalah sebagai berikut:

$$(M11, M12, M13, M14, M21, M31, M32) = (0, 0, 0, 1, 1, 0.65, 0.35)$$

Dengan nilai fungsi tujuan $Z = 13008.20$.

Pada solusi ($M31$ dan $M32$) masih didapatkan nilai fraksional, sehingga *region* 1 perlu dicabang. Pencabangan akan dilakukan pada $M31$ terlebih dahulu. *Region* 1 akan dibagi menjadi 2 *region* baru, yakni untuk $M31 \leq 0$ (*region* 2) dan untuk $M31 \geq 1$ (*region* 3). Pada tiap *region* baru akan ditambahkan 1 batasan baru untuk membatasi daerah solusinya. Misalnya pada *region* 2 batasan-batasannya adalah semua batasan dari *region* 1 ditambah dengan 1 batasan baru, yakni $M31 \leq 0$. Masing-masing *region* baru ini kemudian dipecahkan dengan metode *Linear Programming*, sehingga didapatkan:

- Region 2

  $(M11, M12, M13, M14, M21, M31, M32) = (0, 0.77, 0, 0.23, 1, 0, 1)$

  Dengan nilai fungsi tujuan $Z = 15914.63$.

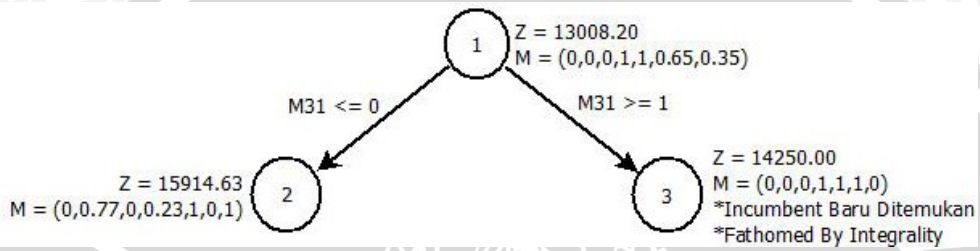
- Region 3

  $(M11, M12, M13, M14, M21, M31, M32) = (0, 0, 0, 1, 1, 1, 0)$

  Dengan nilai fungsi tujuan $Z = 14250.00$.

  Semua variabel dari solusi yang dihasilkan sudah berupa *integer*, sehingga solusi dari *region* ini disimpan sebagai solusi *incumbent*. Ini juga sekaligus menjadikan status dari *region* ini *Fathomed By Integrality* dan tidak perlu ditelusuri lagi.

Sampai saat ini, proses *branch-and-bound* yang terjadi dapat digambarkan dalam bentuk *branch-and-bound tree* pada Gambar 3.2.



Gambar 3.2 *Branch-and-Bound Tree* Pemecahan *Integer Programming* Iterasi-1

3.2.4.2 Iterasi-2

Pada iterasi ini, *region* yang masih aktif dan belum ditelusuri adalah *region* 2. *Region* ini masih memiliki solusi fraksional sehingga perlu dilakukan pencabangan. Kali ini pencabangan dilakukan pada variabel $M12$ dan membagi *region* 2 menjadi *region* $M12 \leq 0$ (*region* 4) dan *region* $M12 \geq 1$ (*region* 5). Sama dengan tahap sebelumnya, masing-masing *region* baru akan memiliki batasan dari *node* di atasnya dan 1 batasan tambahan. Setelah dilakukan pemecahan *Linear Programing* pada masing-masing *node* baru didapatkan:

- Region 4

  $(M11, M12, M13, M14, M21, M31, M32) = (0.33, 0, 0, 0.67, 1, 0, 1)$

  Dengan nilai fungsi tujuan $Z = 15915.69$.

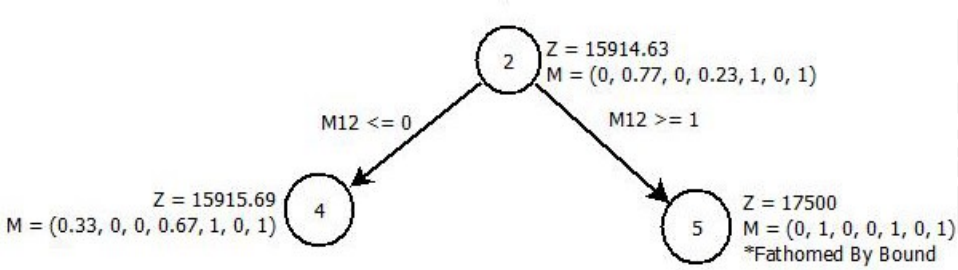
- Region 5

  $(M11, M12, M13, M14, M21, M31, M32) = (0, 1, 0, 0, 1, 0, 1)$

  Dengan nilai fungsi tujuan $Z = 17500.00$.

  Walaupun didapatkan solusi di mana semuanya bernilai *integer*, tetapi nilai fungsi tujuan dari *region* ini tidak lebih baik dari *incumbent solution* (karena kasus yang diselesakan adalah minimasi). Oleh karena itu, *region* ini dikatakan sebagai *Fathomed By Bound* dan tidak akan ditelusuri lagi.

Branch-and-bound tree iterasi ini dapat digambarkan pada Gambar 3.3.



Gambar 3.3 *Branch-and-Bound Tree* Pemecahan *Integer Programming* Iterasi-2

3.2.4.3 Iterasi-3

Pada iterasi ini, *region* 4 merupakan *region* yang masih aktif sehingga akan ditelusuri. Karena masih mengandung solusi fraksional, *region* 4 akan dicabang pada $M11$ menjadi region $M11 \leq 0$ (*region* 6) dan $M11 \geq 1$ (*region* 7). Masing-masing *region* akan diselesaikan menggunakan *Linear Programming* sehingga didapatkan:

- Region 6

  Region ini tidak memiliki solusi *feasible* sehingga *region* ini dikatakan sebagai *Fathomed By Feasibility*.

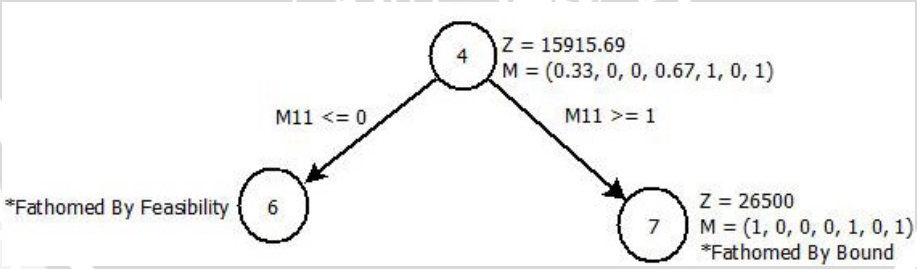
- Region 7

  $(M11, M12, M13, M14, M21, M31, M32) = (1, 0, 0, 0, 1, 0, 1)$

  Dengan nilai fungsi tujuan $Z = 26500.00$.

  Region ini menghasilkan nilai fungsi tujuan yang tidak lebih baik dari *incumbent solution* walaupun semua variabel solusinya *integer*. Oleh karena itu, *region* ini dikatakan *Fathomed By Bound*.

Branch-and-bound tree iterasi ini dapat digambarkan pada Gambar 3.4.

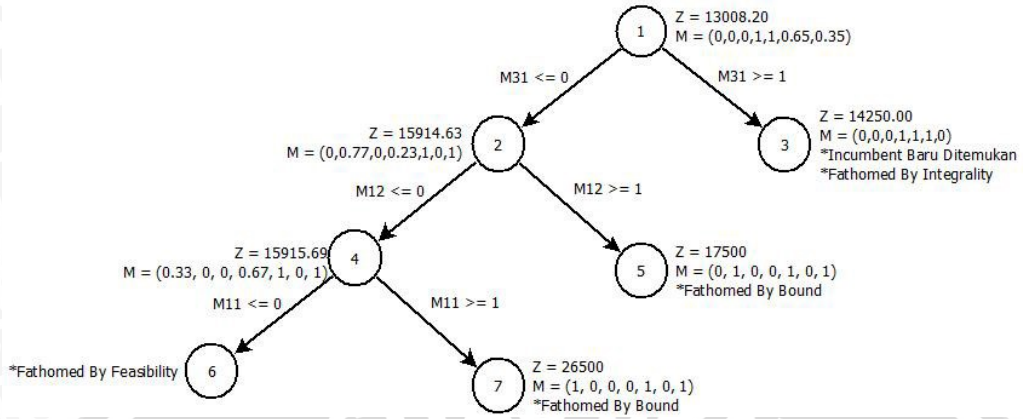


Gambar 3.4 Branch-and-Bound Tree Pemecahan Integer Programming Iterasi-3

Pada akhir iterasi ke-3 ini tidak ditemukan lagi *node* aktif yang dapat ditelusuri sehingga algoritma *branch-and-bound* dihentikan dan solusi *incumbent* diambil sebagai solusi optimal. Dengan demikian, solusi dari *Integer Programming* ini adalah:

$$(M11, M12, M13, M14, M21, M31, M32) = (0, 0, 0, 1, 1, 1, 0)$$

Dengan nilai *objective function* $Z = 14250.00$. Proses *branch-and-bound* secara keseluruhan digambarkan pada Gambar 3.5.



Gambar 3.5 *Branch-and-Bound Tree* Pemecahan Integer Programming Keseluruhan

Solusi dari *Integer Programming* tersebut 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

3.3 Pengumpulan data

Dalam penelitian ini, data yang dibutuhkan dibagi menjadi lima kelompok dengan metode pengumpulan yang berbeda-beda. Beberapa data juga memerlukan validasi dari pakar gizi untuk menambah kekuatannya sebagai data penelitian. Bagian ini akan memberikan uraian mengenai data-data yang digunakan serta metode pengumpulannya.

3.3.1 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/. Pada penelitian ini akan digunakan *tool* yang dikembangkan khusus untuk melakukan pencarian dan transfer data dari basis data USDA ke basis data lokal. Dengan memanfaatkan API *(Application Program Interface)* yang disediakan oleh USDA, *tool* ini akan mempermudah pencarian data kandungan gizi bahan makanan di antara ribuan data yang ada pada basis data USDA serta menyalinnya ke basis data lokal. Untuk keperluan penelitian ini, nama bahan makanan akan disesuaikan dari

nama internasional menjadi nama Indonesia. Data bahan-bahan yang telah siap digunakan akan disertakan pada bagian lampiran.

Bahan-bahan berupa bumbu (misalnya merica, pala dan lain-lain), bahan tambahan atau pelengkap (misalnya bawang goreng), atau bahan-bahan lain (misalnya es batu) yang kontribusinya pada nilai gizi dianggap tidak signifikan diberi nilai 0 untuk setiap nutrien. Dengan kata lain, bahan-bahan ini tidak akan diperhitungkan dalam perhitungan nilai gizi suatu menu. Tabel 3.13 adalah contoh dari data kandungan gizi bahan makanan.

Tabel 3.13 Contoh data kandungan gizi bahan makanan

id	tipe	nama	energy	protein	vitamin_a	vitamin_b1	vitamin_c
1001	utama	**Mentega** *Butter, salted*	717	0.85	684	0.005	0
1009	utama	**Keju Cheddar** *Cheese, cheddar*	404	22.87	330	0.029	0
1	bumbu	**Kencur**	0	0	0	0	0
2	bumbu	**Lengkuas**	0	0	0	0	0

3.3.2 Data satuan bahan makanan

Data satuan bahan makanan adalah data yang berisi satuan-satuan atau unit-unit dari sebuah bahan makanan. Contoh satuan atau unit 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. Data satuan bahan makanan yang telah siap digunakan akan disertakan pada bagian lampiran. Tabel 3.14 adalah contoh data satuan bahan makanan.

Tabel 3.14 Contoh data satuan bahan makanan

Id	Nama	Satuan
1001	**Mentega** *Butter, salted*	Sendok Makan (.sdm) *Eqv. 14.2 gram* Sendok Teh (.sdt) *Eqv. 4.73 gram*
1009	**Keju Cheddar** *Cheese, cheddar*	Iris (1 ons) (.iris) *Eqv. 28 gram*
1084	**Susu Cair** *Milk, lowfat, fluid, 1% milkfat, protein fortified, with added vitamin A and vitamin D*	Mililiter (.ml) *Eqv. 1gram* Cubic Centimeter (.cc) *Eqv. 1gram*
1090	**Susu Bubuk** *Milk, dry, whole, with added vitamin D*	Bungkus (.bks) *Eqv. 27 gram*
1095	**Susu Kental Manis** *Milk, canned, condensed, sweetened*	Sendok Makan (.sdm) *Eqv. 15 gram* Sachet (.sachet) *Eqv. 40 gram*

3.3.3 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. Untuk keperluan ini, penulis mendapatkan bantuan dari dosen Program Studi Ilmu Gizi Fakultas Kedokteran Universitas Brawijaya, Ibu Titis Sari Kusuma (Kusuma, 2016). Data menu yang telah divalidasi oleh pakar dan siap digunakan akan disertakan pada bagian lampiran. Contoh dari data menu dapat digambarkan dalam bentuk tabel pada Tabel 3.15.

Tabel 3.15 Contoh data menu

Id	Nama	Porsi	Porsi Penyajian Standar	Harga Fuzzy (s,l,r)	Bahan Utama	Bahan Tambahan / Bumbu
1	Sambal Goreng Ampela	5	1	(5700, 1000, 500)	- Ampela Ayam : 1 .kg *Fuzzy eqv. (1000, 0, 0) gram* - Wortel : 2 .buah *Fuzzy eqv. (122, 5, 5) gram* - Kentang : 2 .buah *Fuzzy eqv. (426, 10, 10) gram*	- Kencur : secukupnya - Lengkuas : secukupnya - Penyedap Rasa : secukupnya - Minyak Goreng : secukupnya - Kemiri : 2 .btr - Daun Salam : 1 .lbr - Garam : secukupnya - Bawang Putih : 2 .siung - Jahe : secukupnya - Bawang Merah : 3 .siung - Cabe Merah : 4 .buah - Cabe Merah : 2 .buah - Santan : 0 .gr
2	Nasi	1	1,1/2	(1700, 500, 500)	- Nasi : 1 .prg *Fuzzy eqv. (200, 20, 10) gram*	
3	Oseng-oseng Kacang Panjang Campur Tempe	4	1,1/2	(1400, 500, 1000)	- Saus Tiram : 2 .sdm *Fuzzy eqv. (36, 2, 2) gram* - Kacang Panjang : 100 .gr *Fuzzy eqv. (100, 0, 0) gram* - Tempe : 1 .buah *Fuzzy eqv. (200, 10, 50) gram* - Kecap : 2 .sdm *Fuzzy eqv. (36, 2, 2) gram*	- Minyak Goreng : secukupnya - Air : secukupnya - Merica : secukupnya - Daun Salam : 2 .lbr - Garam : secukupnya - Bawang Putih : 2 .siung - Jahe : 2 .ruas jari - Bawang Merah : 2 .siung - Bawang Bombay : 1 .buah - Cabe Rawit : 7 .buah - Cabe Merah : 1 .buah

3.3.4 Data menu set

Data ini digunakan untuk menguji optimalitas algoritma. Data berisi susunan menu-menu yang mungkin dikonsumsi seseorang dalam satu hari. Susunan menu-menu tersebut tidak harus diperkirakan dapat memenuhi kebutuhan gizi

seseorang dalam satu hari. Dengan kata lain, tidak perlu diketahui secara *a priori* bahwa semua menu set tersebut *feasible*. Justru data-data 'menyimpang' ini dapat digunakan untuk menguji apakah algoritma dapat mengenali kondisi di mana memang tidak ada solusi yang optimal *(no feasible solution)*.

Sebelum digunakan sebagai data pengujian, data menu set ini akan terlebih dahulu dikonsultasikan dengan pakar gizi, yakni Ibu Titis Sari Kusuma (Kusuma, 2016), untuk mendapatkan validasi bahwa kasus-kasus yang direpresentasikan dalam data memang mungkin terjadi dan data dianggap layak dijadikan sebagai data uji. Data Menu Set yang telah divalidasi oleh pakar dan siap digunakan akan disertakan pada bagian lampiran. Contoh dari data Menu Set adalah seperti pada Tabel 3.16.

Tabel 3.16 Contoh data menu set

Id	Nama	Menu
1	Menu Set 01	- Sambal Goreng Ampela - Nasi - Oseng-oseng Kacang Panjang Campur Tempe - Nasi - Brownis Pisang - Jus Wortel Tomat
2	Menu Set 02	- Nasi Goreng Kecap Orak Arik Telor - Susu Kental Manis - Nasi - Sayur Asem
3	Menu Set 03	- Nasi - Telur Dadar Jamur - Susu Kental Manis - Lodeh Rebung - Nasi - Ayam Lada Hitam - Nasi

3.3.5 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 *Dietary Reference Intakes* (DRI) tahun 2001. DRI dipilih untuk dijadikan acuan dibandingkan dengan AKG karena DRI memberikan data yang jelas mengenai batas atas *(Upper Level)* sehingga dapat digunakan untuk menguji pengaruh batas atas. Selain itu DRI juga menyediakan rumus *Estimated Energy Requirement* (EER)

untuk menghitung kebutuhan kalori individu. Data lengkap DRI akan disertakan pada bagian lampiran.

3.4 Strategi pengujian

Berdasarkan pada tujuan penelitian, pengujian akan dilakukan dalam empat bentuk, dengan kode "Pengujian A" hingga "Pengujian D" yang masing-masing strateginya akan dibahas pada subbab selanjutnya. Nutrien-nutrien yang diikutsertakan dalam tiap pengujian adalah: Energi, Protein, Vitamin A, Vitamin B1 dan Vitamin C.

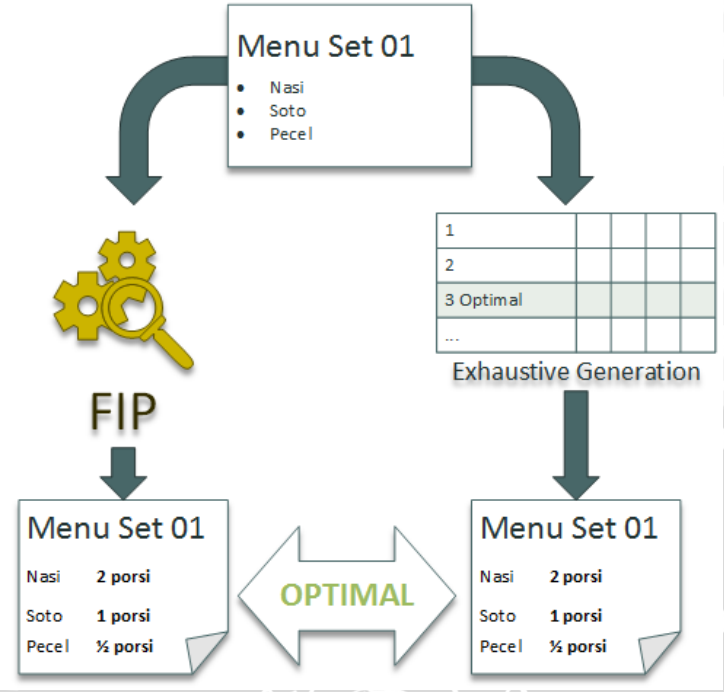
Setiap pengujian diasumsikan dilakukan untuk subjek yang sama. Dengan demikian, profil subjek dianggap tetap. Subjek diasumsikan adalah laki-laki berusia 20 tahun dengan tinggi badan 163 cm, berat badan 60 kg dan memiliki tingkat aktivitas sedang. Dari Tabel 2.3 didapatkan nilai PA subjek adalah 1.11, dan dengan menggunakan rumus dari Tabel 2.2 didapatkan nilai EER subjek adalah 2507.3 *Kcal*. Berdasarkan data RDA kebutuhan gizi subjek dapat diuraikan pada Tabel 3.17. Anggaran yang dimiliki oleh subjek diasumsikan sebesar Rp 50.000.

Tabel 3.17 Kebutuhan gizi subyek

Nutrien	Min.	Maks.
Energi	2,507.3	Infinity
Protein	56	Infinity
Vitamin A	900	3,000
Vitamin B1	1.2	Infinity
Vitamin C	90	2,000

3.4.1 Ukuran kualitas algoritma

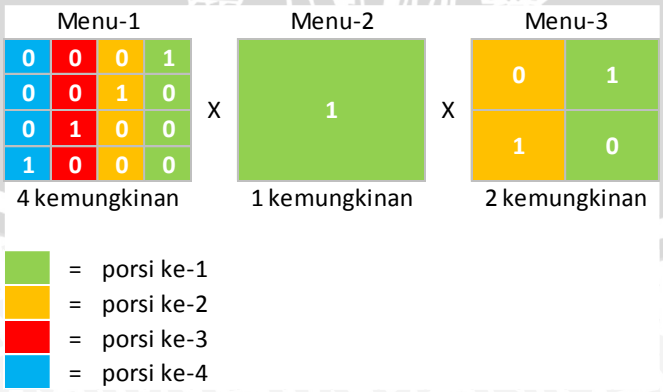
Ukuran yang digunakan untuk menentukan kualitas algoritma FIP dalam menyelesaikan permasalahan optimasi kandungan gizi menu harian adalah tingkat optimalitasnya. Uji optimalitas dilakukan dengan membandingkan kombinasi hasil optimasi algoritma FIP dengan semua kemungkinan kombinasi yang dibangkitkan secara menyeluruh *(exhaustive)*. Untuk setiap set menu harian, jika hasil optimasi dengan menggunakan algoritma FIP sama dengan kombinasi paling optimal berdasarkan pembangkitan *exhaustive*, maka untuk set tersebut algoritma dikatakan telah optimal. Kombinasi paling optimal berdasarkan pembangkitan *exhaustive* adalah kombinasi yang memiliki harga paling murah dan kandungan gizinya ada dalam batasan-batasan yang telah ditentukan oleh acuan kecukupan gizi. Gambar 3.6 adalah gambaran dari proses uji optimalitas algoritma untuk suatu *Menu Set*.



Gambar 3.6 Diagram penentuan optimalitas algoritma untuk suatu Menu Set

3.4.2 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. Agar lebih memudahkan, kombinasi-kombinasi tersebut diubah ke dalam bentuk sebuah kode alamat *(address)*. Penentuan alamat kombinasi diilustrasikan pada Gambar 3.7 dan dicontohkan pada Tabel 3.18. Dari contoh tersebut, dapat dihitung bahwa akan ada $4 \times 1 \times 2 = 8$ kemungkinan kombinasi.



Gambar 3.7 Ilustrasi penentuan alamat kombinasi

Tabel 3.18 Contoh penentuan alamat kombinasi

Soto Ayam				Nasi	Jus Alpokat		Address
2	1	1/2	1/4	1	1	1/2	
			✔	✔		✔	0001.1.01
		✔		✔		✔	0010.1.01
dst.				dst.	dst.		dst.

Selanjutnya, setiap kombinasi akan dihitung nilai gizinya dengan cara yang sama seperti pada contoh perhitungan manual (Tabel 3.4, Tabel 3.5 dan Tabel 3.6), hanya saja yang diperlukan pada tahap pengujian ini bukan *fuzzy number* (s, l, r), melainkan nilai *lower* $(s - l)$ dan *upper* $(s + r)$ *fuzzy number* tersebut. Harga dari kombinasi juga dihitung dalam tahap ini. Perhitungan tersebut dapat dicontohkan seperti pada Tabel 3.19 dan Tabel 3.20.

Tabel 3.19 Contoh perhitungan nilai gizi tiap kemungkinan (1)

Address	energy	protein	vitamin_a
	850 - Infinity	21.25 - Infinity	382.5 - Infinity
0001.1.01	1,002.3 - 1,709.4	15.94 - 26.48	1,447.69 - 1,925.44
0001.1.10	1,695.2 - 2,829.4	24.17 - 38.93	2,641.69 - 3,567.94
0010.1.01	1,047.7 - 1,770.8	18.26 - 29.71	1,701.38 - 2,208.38
0010.1.10	1,740.6 - 2,890.8	26.49 - 42.16	2,895.38 - 3,850.88
0100.1.01	1,138.5 - 1,893.6	22.88 - 36.17	2,208.75 - 2,774.25
0100.1.10	1,831.4 - 3,013.6	31.11 - 48.62	3,402.75 - 4,416.75
1000.1.01	1,320.1 - 2,139.2	32.14 - 49.09	3,223.5 - 3,906
1000.1.10	2,013 - 3,259.2	40.37 - 61.54	4,417.5 - 5,548.5

Tabel 3.20 Contoh perhitungan nilai gizi tiap kemungkinan (2)

Address	vitamin_b1	vitamin_c	price
	0.43 - Infinity	38.25 - Infinity	0 - 30000
0001.1.01	0.48 - 0.76	77.06 - 97.47	8,125 - 9,875
0001.1.10	0.78 - 1.14	148.96 - 189.22	10,875 - 13,125
0010.1.01	0.52 - 0.83	82.21 - 103.19	10,000 - 12,000
0010.1.10	0.81 - 1.21	154.11 - 194.94	12,750 - 15,250
0100.1.01	0.59 - 0.97	92.53 - 114.63	13,750 - 16,250
0100.1.10	0.89 - 1.35	164.43 - 206.38	16,500 - 19,500
1000.1.01	0.74 - 1.25	113.15 - 137.5	21,250 - 24,750
1000.1.10	1.03 - 1.63	185.05 - 229.25	24,000 - 28,000

Selanjutnya, kombinasi-kombinasi yang tidak memenuhi kriteria *(infeasible)* akan disingkirkan dari daftar. Pada contoh ini, kombinasi 0001.1.01 memiliki nilai protein kurang dari batas bawah yang diperbolehkan, sehingga dikatakan

infeasible. Begitu pula dengan kombinasi 0010.1.01. Setelah daftar hanya berisi kombinasi yang memenuhi kebutuhan gizi *(feasible)*, dapat diketahui kombinasi optimal, yakni kombinasi yang memiliki harga paling murah. Pada contoh ini, kombinasi optimal adalah 0001.1.10. Jika diuraikan, kombinasi tersebut adalah:

- Soto Ayam : ¼ porsi
- Nasi : 1 porsi
- Jus Alpokat : 1 porsi.

Jika kombinasi ini sama dengan kombinasi yang diberikan oleh FIP, maka untuk kasus ini algoritma FIP dikatakan telah optimal. Tabel 3.21 merupakan contoh dari tabel yang akan digunakan untuk melakukan pengujian optimalitas.

Tabel 3.21 Contoh tabel 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 min	15.94 - 26.48 min	1447.09 - 1,925.44 min	0.48 - 0.76 min	77.06 - 97.47 min	8,125 - 9.875 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	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 max	40.37 - 61.54 max	4,417.5 - 5,548.5 max	1.03 - 1.63 max	185.05 - 229.25 max	24,000 - 28,000 max

Legenda

	Optimal berdasarkan FIP
max	Nilai maksimum
min	Nilai minimum
	Melebihi batas bawah
	Melebihi batas atas
teks	Kombinasi *feasible*
teks	Kombinasi *infeasible*
teks	Kombinasi optimal

3.4.3 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.

3.4.3.1 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. Tabel 3.22, Tabel 3.23, dan Tabel 3.24 merupakan gambaran data dari beberapa nilai parameter ini.

Tabel 3.22 Menu dengan *menu food weights fuzziness* = 'bebas'

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

Tabel 3.23 Menu dengan *menu food weights fuzziness* = 0%

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

Tabel 3.24 Menu dengan *menu food weights fuzziness* = 50%

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

3.4.3.2 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. Tabel 3.25, Tabel 3.26, dan Tabel 3.27 merupakan gambaran data menu dari beberapa nilai parameter ini.

Tabel 3.25 Harga menu dengan *menu price fuzziness* = 'bebas'

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

Tabel 3.26 Harga menu dengan *menu price fuzziness* = 0%

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

Tabel 3.27 Harga menu dengan *menu price fuzziness* = 50%

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

3.4.3.3 Keburaman batasan gizi (reference fuzziness)

Parameter ini menggambarkan persen toleransi terhadap batasan-batasan gizi yang telah didefinisikan melalui angka-angka referensi (EER, RDA/AI, UL). 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. Tabel 3.28 dan Tabel 3.29 merupakan gambaran data batasan dengan nilai *reference fuzziness* sebesar 0% dan 10%.

Tabel 3.28 Batasan dengan *reference fuzziness* = 0%

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

Tabel 3.29 Batasan dengan *reference fuzziness* = 10%

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

3.4.4 Pengujian pengaruh keburaman berat bahan penyusun menu terhadap optimalitas (pengujian A)

Pengujian ini digunakan untuk mengetahui pengaruh keburaman data berat bahan-bahan penyusun menu yang dimasukkan oleh pengguna terhadap tingkat optimalitas algoritma. Pengujian ini memiliki kode "Pengujian A". 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). Kasus-kasus yang tak memiliki solusi *(no feasible solution)* tidak diperhitungkan. Tabel 3.30 merupakan data parameter yang digunakan untuk skenario-skenario pada "Pengujian A".

3.4.5 Pengujian pengaruh keburaman harga menu terhadap optimalitas (pengujian B)

Pengujian ini digunakan untuk mengetahui pengaruh keburaman harga menu yang dimasukkan oleh pengguna terhadap tingkat optimalitas algoritma. Pengujian dengan kode "Pengujian B" ini akan menguji 50 buah menu set dalam beberapa skenario berbeda. 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". Tabel 3.31 merupakan data parameter yang digunakan untuk skenario-skenario pada "Pengujian B".

3.4.6 Pengujian pengaruh keburaman batasan pada optimalitas (pengujian C)

Pengujian dengan kode "Pengujian C" 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". Tabel 3.32 merupakan data parameter yang digunakan untuk skenario-skenario pada "Pengujian C".

3.4.7 Pengujian optimalitas untuk kasus nyata (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. Tabel 3.33 menunjukkan parameter-parameter yang digunakan dalam pengujian ini.

Tabel 3.30 Parameter dari skenario-skenario "pengujian A"

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 3.31 Parameter dari skenario-skenario "pengujian B"

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 3.32 Parameter dari skenario-skenario " pengujian C"

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 3.33 Parameter dari skenario-skenario " pengujian D"

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

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini akan didahului dengan pembahasan mengenai arsitektur, spesifikasi, dan kebutuhan sistem yang akan dibangun. Pembahasan dilanjutkan dengan uraian mengenai rancangan komponen-komponen utama sistem, rancangan basis data, serta rancangan antarmuka pengguna. Pada bagian akhir dari bab ini akan dibahas mengenai implementasi algoritma serta implementasi antarmuka pengguna hingga dapat dihasilkan sebuah sistem yang siap uji.

4.1 Arsitektur sistem

Untuk dapat menerapkan serta menguji algoritma *Fuzzy Integer Programming* untuk optimasi kandungan gizi menu harian diperlukan pengembangan sebuah sistem berupa perangkat lunak atau aplikasi. Sistem ini digunakan untuk membantu proses pemasukan data, penyuntingan data (jika diperlukan) serta menyajikan antarmuka untuk pengujian.

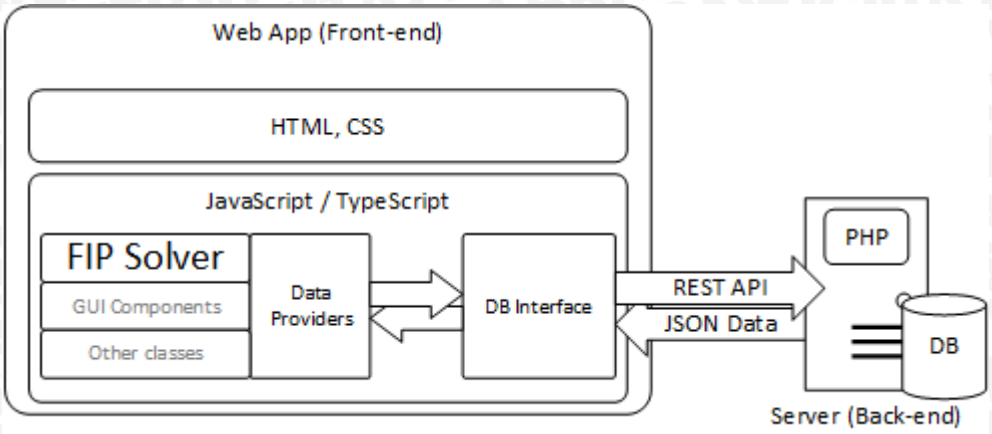
Aplikasi menerima masukan berupa sebuah *menu set*, yakni daftar menu-menu yang ingin dioptimasi, melalui antarmuka pengguna. Aplikasi sebenarnya juga membutuhkan data profil subjek seperti tinggi badan berat badan dan lain-lain. Namun seperti yang telah disebutkan dalam strategi pengujian, data-data ini akan diberi nilai tetap. Setelah mendapatkan masukan, aplikasi kemudian akan melakukan permintaan *(request)* ke *server* basis data melalui sebuah *Application Program Interface* (API). Tergantung dari *request*-nya, *server* akan memberikan balasan berupa data:

- Kandungan gizi tiap menu dalam *menu set*
- Batasan RDA subjek
- Batasan UL subjek

Setelah data-data tersebut didapatkan, aplikasi akan memanggil kelas yang mengimplementasikan algoritma *Fuzzy Integer Programming* untuk menghitung kombinasi porsi optimal untuk tiap menu. Keluaran dari aplikasi ini adalah kombinasi porsi paling optimal dari tiap menu dalam *menu set* masukan. Selain itu, aplikasi juga memiliki antarmuka untuk melakukan pengujian algoritma.

Algoritma akan diimplementasikan dalam bentuk sebuah aplikasi *web (web app)*. Hal ini didasarkan atas pertimbangan bahwa antarmuka *web* dirasa lebih intuitif dan ramah pengguna *(user-friendly)*. Selain itu, informasi dan data yang perlu ditampilkan dapat ditampilkan dengan antarmuka yang lebih menarik.

Aplikasi terbagi menjadi dua sisi, sisi klien *(client side* atau *front-end)* dan sisi *server (server side* atau *back-end)*. Bagian *front-end* akan berkenaan dengan interaksi pengguna dan penampilan data sedangkan bagian *back-end* berkenaan dengan operasional basis data. Arsitektur aplikasi secara umum dapat digambarkan seperti pada Gambar 4.1.



Gambar 4.1 Arsitektur aplikasi

4.2 Spesifikasi dan kebutuhan sistem

Agar dapat dibangun sebuah sistem sesuai dengan arsitektur yang telah direncanakan pada bagian "Arsitektur Sistem", dibutuhkan spesifikasi-spesifikasi sistem sebagai berikut:

- ***Front-end***

  Bahasa : HTML, CSS, JavaScript / TypeScript

  JavaScript Framework : Angular 2 (v2.0.0-beta.3)

  CSS Framework : Bootstrap (v3.3.6)

- ***Back-end***

  Bahasa : PHP (v5.5.15)

  RDBMS : MySQL (v5.6.20)

Sebagai tambahan, berikut adalah gambaran lingkungan pengembangan yang akan digunakan:

- **Perangkat keras**

  CPU : Intel® Core™ 2 Quad Q6600 @ 2.8 GHz *(O.C)*

  RAM : 5 GB DDR 3, 640.2 MHz *(O.C) Dual Channel*

  HDD : 1TB SATA

- **Perangkat lunak**

  Sistem Operasi : Windows 7 Ultimate SP1, 64 Bit Edition

  Web Server : XAMPP (v3.2.1)

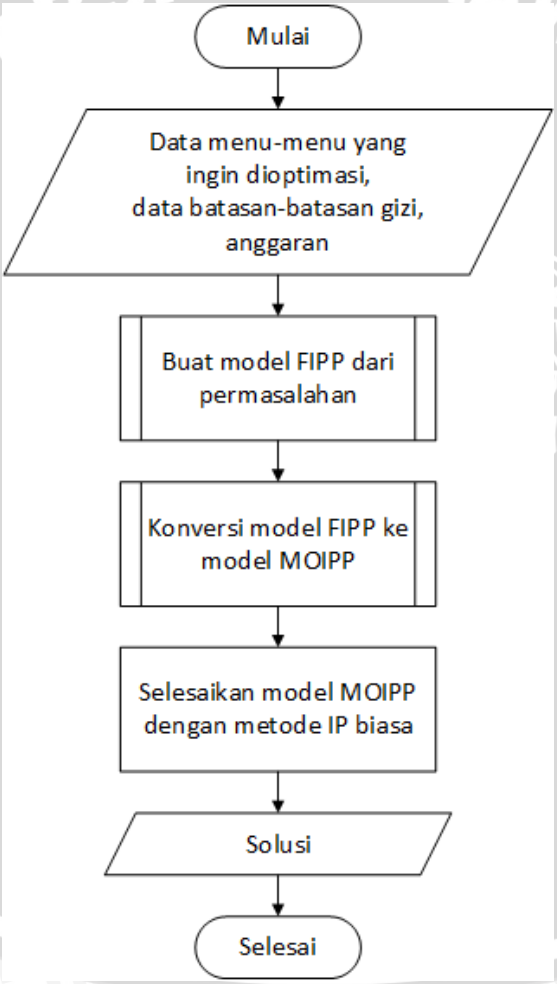
  IDE : 1. JetBrains WebStorm 11.0

  2. JetBrains PHPStorm 10.0

Web Browser	:	1. Mozilla Firefox Developer Edition (v44.0a2) 2. Google Chrome (v49.0.2623.112 m)
NodeJS	:	v5.7.1
NPM	:	v3.6.0
Version Control	:	Git (v2.7.2.windows.1)

4.3 Rancangan Algoritma

Rancangan dari algoritma optimasi kandungan gizi menu harian dengan menggunakan *Fuzzy Integer Programming* dapat digambarkan dalam bentuk diagram alir *(flowchart)* seperti pada Gambar 4.2.

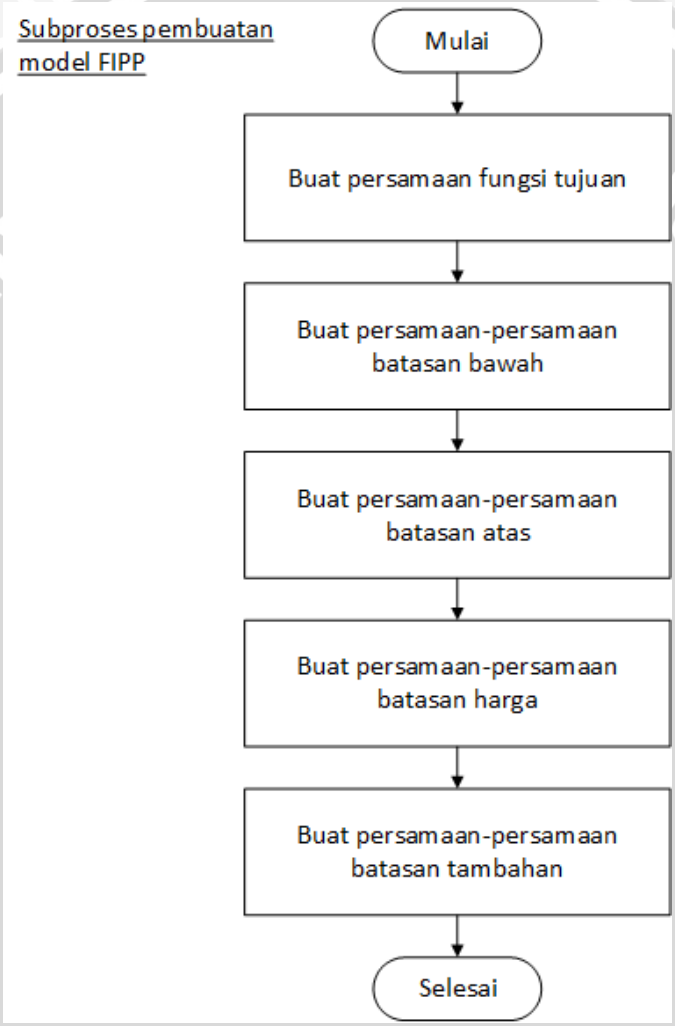


Gambar 4.2 *Flowchart* algoritma

Algoritma diawali dengan menerima masukan berupa data menu-menu yang ingin dioptimasi, data batasan-batasan gizi, serta data batasan biaya atau anggaran. Dari data-data ini dibuat sebuah model matematika *Fuzzy Integer Programming Problem* (FIPP). Model FIPP yang telah terbentuk kemudian

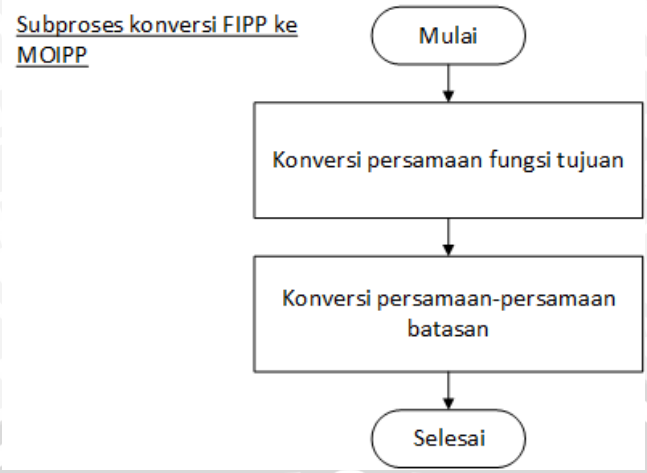
dikonversi menjadi model matematika *Multiple Objective Integer Programming Problem* (MOIPP). Model ini selanjutnya dapat diselesaikan dengan metode *Integer Programming* biasa sehingga dapat diperoleh sebuah solusi.

Pada bagian pembuatan model FIPP, algoritma akan membuat persamaan fungsi tujuan dan persamaan-persamaan batasan. Seperti yang telah dijelaskan pada bab 3.2.2, persamaan batasan terdiri atas batasan-batasan bawah, batasan-batasan atas, batasan anggaran, serta beberapa batasan tambahan. Pembuatan model FIPP ini dapat digambarkan dalam bentuk *flowchart* seperti pada Gambar 4.3.



Gambar 4.3 *Flowchart* algoritma, subproses pembuatan model FIPP

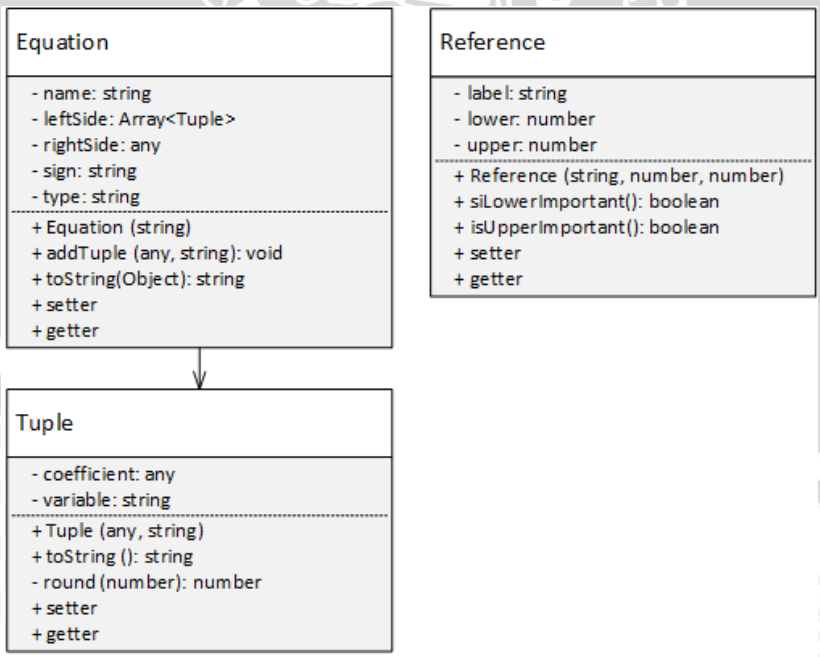
Pada bagian konversi model FIPP ke model MOIPP, algoritma akan melakukan konversi dengan metode seperti yang telah dibahas pada bab 3.2.3. Bagian ini secara sederhana dapat digambarkan dalam bentuk *flowchart* seperti pada Gambar 4.4.



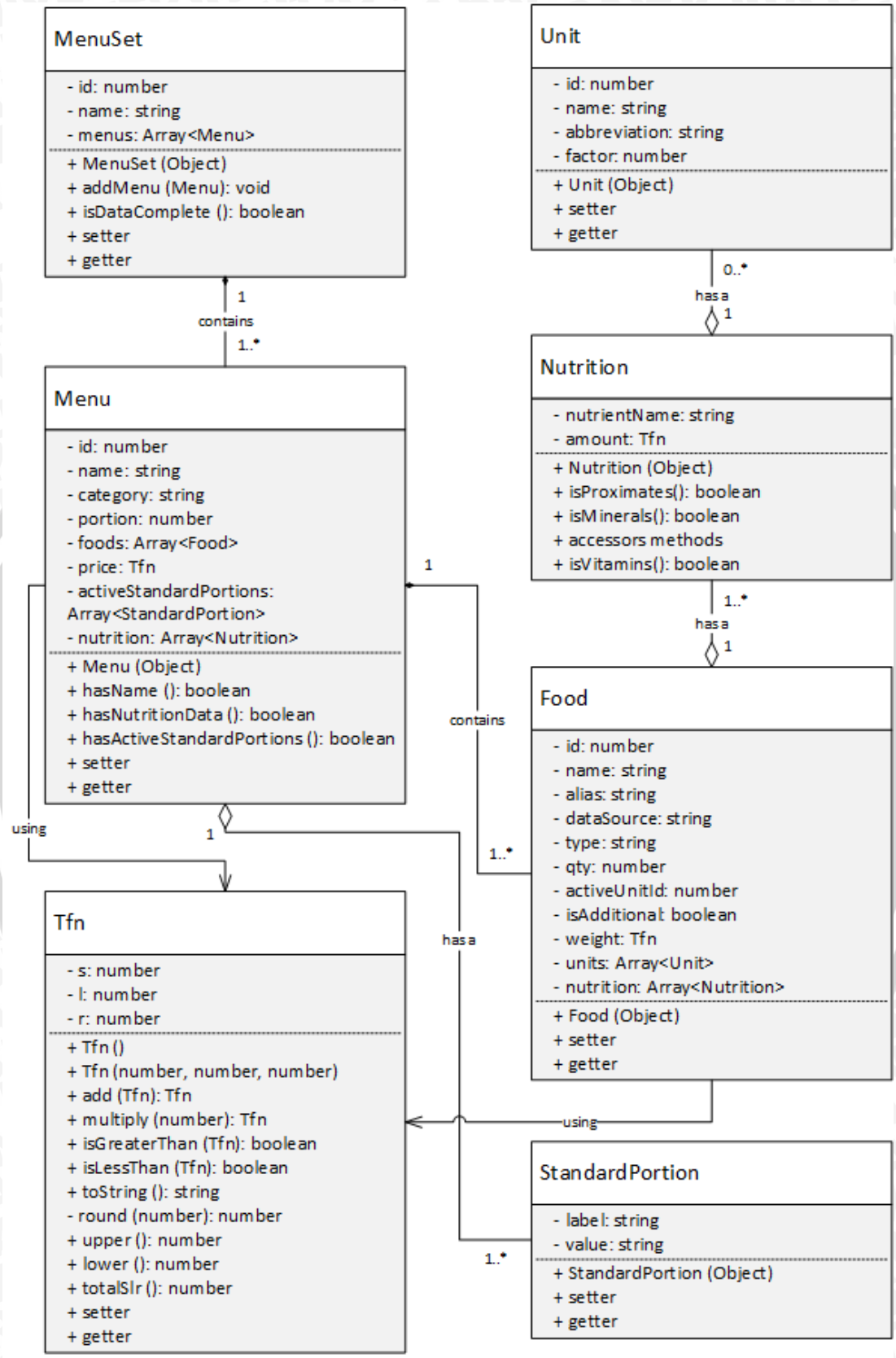
Gambar 4.4 *Flowchart* algoritma, subproses konversi FIPP ke MOIPP

4.4 Rancangan kelas-kelas utama

Pada bagian ini akan dijelaskan mengenai rancangan struktur dari kelas-kelas utama yang akan digunakan pada tahap implementasi. Pada penerapannya akan sangat mungkin ada tambahan kelas-kelas lain yang secara spesifik menangani tugas tertentu. Misalnya, komponen antarmuka pengguna *(User Interface Component)*, antarmuka basis data *(DB Interface)*, penyedia data *(Data Provider)* dan kelas-kelas lain. Namun karena kelas-kelas ini tidak terlalu relevan dengan fokus utama penelitian, kelas-kelas ini tidak akan dibahas. Rancangan kelas-kelas utama dalam penelitian ini dapat digambarkan dalam bentuk *class diagram* pada Gambar 4.5 dan Gambar 4.6.



Gambar 4.5 *Class diagram* dari kelas-kelas utama (a)



Gambar 4.6 *Class diagram* dari kelas-kelas utama (b)

4.4.1 Kelas Tfn *(Triangular Fuzzy Number)*

Kelas ini digunakan untuk merepresentasikan sebuah bilangan *fuzzy triangular (Triangular Fuzzy Number).* Kelas Tfn memiliki spesifikasi sebagai berikut:

- Memiliki properti *private* s, l dan r, masing-masing mewakili nilai *mid*, *left spread* dan *right spread* sebuah *triangular fuzzy number*.
- Dapat diinisialisasi dengan dua *constructor*. Satu *constructor* tanpa argumen yang akan memberikan nilai 0 untuk properti s, l dan r. *Constructor* lainnya menerima objek sebagai argumen yang akan memberikan nilai untuk properti s, l, dan r sesuai dengan nilai yang di-*passing* melalui objek.
- Memiliki fungsi *add*, *multiply*, *isGreaterThan* dan *isLessThan*. Masing-masing fungsi secara berurutan berguna untuk: menambahkan obyek Tfn lain ke Tfn, mengalikan Tfn dengan sebuah pengali *(multiplier)*, memeriksa apakah Tfn lebih besar dari Tfn lain dan memeriksa apakah Tfn lebih kecil dari Tfn lain.
- Memiliki fungsi *toString* untuk mendapatkan representasi *string* dari Tfn. Fungsi ini juga memerlukan fungsi *round* yang akan membulatkan angka jika diperlukan representasi yang tidak terlalu panjang, misalnya ketika Tfn ditampilkan di antarmuka.
- Memiliki *setter* dan *getter* untuk mengakses properti *private* s, l, dan r.
- Memiliki *getter upper*, *lower* dan *totalSlr* yang masing-masing mengembalikan nilai $(s + r)$, $(s - l)$ dan $(s + l + r)$.

4.4.2 Kelas Equation dan Tuple

Kelas *Equation* digunakan untuk merepresentasikan sebuah persamaan, sedangkan kelas *Tuple* digunakan untuk merepresentasikan pasangan koefisien dan variabel dalam sebuah persamaan. Sebuah kelas *Tuple* memiliki spesifikasi sebagai berikut:

- Memiliki properti *private coefficient* dan *variable*. *Coefficient* dapat berupa *string*, *number* atau Tfn. Sedangkan variable berupa *string*.
- Diinisialisasi dengan satu *constructor* yang menerima dua argumen, *coefficient* dan *variable*.
- Memiliki fungsi *toString* untuk mendapatkan representasi *string* dari pasangan koefisien dan variabel. Juga memiliki fungsi *round* untuk membulatkan angka jika diperlukan.
- Memiliki *setter* dan *getter* untuk mengakses properti *private coefficient* dan *variable*.

Sebuah kelas *Equation* memiliki spesifikasi sebagai berikut:

- Memiliki properti *name*, *leftSide*, *rightSide*, *sign* dan *type* yang merepresentasikan nama persamaan, ruas kiri persamaan, ruas kanan persamaan, tanda persamaan dan tipe persamaan.
- Properti *leftSide* merupakan *array* yang berisi beberapa *Tuple*.
- Memiliki *constructor* yang secara *default* memberikan nilai 'Z' untuk *rightSide* dan '=' untuk *sign*.
- Memiliki fungsi *addTuple* yang dapat digunakan untuk menambahkan *Tuple* baru ke ruas kiri persamaan. Fungsi ini menerima dua argumen, *coefficient* dan *variable*.
- Memiliki fungsi *toString* untuk mendapatkan representasi persamaan dalam bentuk *string*. Fungsi ini menerima argumen berupa format dan mengembalikan representasi *string* dari persamaan sesuai format yang di-*passing* melalui argumen.
- Memiliki *setter* dan *getter* untuk mengakses setiap properti *private*.

4.4.3 Kelas Nutrition

Kelas ini merepresentasikan suatu nutrien beserta nilainya untuk suatu bahan tertentu. Kelas *Nutrition* memiliki spesifikasi sebagai berikut:

- Memiliki properti *private nutrientName* dan *amount*, masing-masing merepresentasikan nama nutrien dan nilai dari nutrien tersebut. *nutrientName adalah* sebuah *string*, sedangkan *amount* adalah sebuah Tfn.
- Diinisialisasi dengan satu *constructor* yang menerima argumen berupa objek. Nilai properti *private nutrientName* dan *amount* akan diset sesuai dengan properti dari objek tersebut.
- Memiliki fungsi *isProximates*, *isMinerals* dan *isVitamins* yang masing-masing dapat digunakan untuk mengetahui jenis nutrien berdasarkan nama nutrien.
- Memiliki *setter* dan *getter* untuk mengakses setiap properti *private*.

4.4.4 Kelas Reference

Kelas ini merepresentasikan suatu nilai referensi yang memberikan batasan pada suatu persamaan. Kelas *Reference* bisa digunakan untuk mendefinisikan batasan RDA, UL atau batasan harga. Misalnya, untuk mendefinisikan bahwa nutrien "protein" hanya boleh berada antara 25 hingga 50 maka perlu dibuat sebuah objek *Reference* dengan properti *label* adalah "protein", properti *lower* diberi nilai 25 dan properti *upper* diberi nilai 50. Untuk merepresentasikan batasan

harga *Reference* harus memiliki label *'price'*. Dengan demikian, kelas *reference* memiliki spesifikasi:

- Memiliki properti *private label, lower* dan *upper*, masing-masing merepresentasikan nama, batas bawah dan batas atas *Reference*.
- Diinisialisasi dengan satu *constructor* yang menerima tiga argumen yakni *label*, *lower* dan *upper* yang masing-masing akan digunakan untuk memberikan nilai pada properti *private* yang bersesuaian milik kelas ini.
- Memiliki *setter* dan *getter* untuk mengakses setiap properti *private*.
- Memiliki *getter isLowerImportant* yang akan mengembalikan *boolean true* jika nilai *lower* tidak *null* dan nilai *lower* tidak sama dengan 0.
- Memiliki *getter isUpperImportant* yang akan mengembalikan *boolean true* jika nilai *upper* tidak *null* dan nilai *upper* tidak sama dengan *Infinity*.

Getter isLowerImportant dan *isUpperImportant* nantinya diperlukan dalam pembuatan *constraint*. Jika sebuah *reference* memiliki *isLowerImportant* bernilai *true* maka nilai *lower*-nya perlu dijadikan sebuah batasan bawah *(lower constraint)*. Demikian pula jika sebuah *reference* memiliki *isUpperImportant* bernilai *true* maka nilai *upper*-nya perlu dijadikan sebuah batasan atas *(upper constraint)*. *Getter* terakhir adalah *isPriceReference*, yang jika bernilai *true* maka *reference* tersebut adalah *reference* untuk membatasi harga. *Reference* dengan tipe ini akan diperlakukan secara khusus pada saat pembuatan persamaan.

4.4.5 Kelas Unit

Kelas ini mewakili unit satuan suatu bahan tertentu beserta faktor konversinya (dalam gram). Kelas *Unit* memiliki spesifikasi sebagai berikut:

- Memiliki properti *private id*, *name*, *abbreviation* dan *factor*, masing-masing merepresentasikan *id* unit, nama unit, singkatan unit serta faktor konversi unit (berat 1 unit dalam gram).
- Diinisialisasi dengan satu *constructor* yang menerima argumen berupa obyek. Nilai properti-properti *private* diset sesuai dengan properti dari obyek tersebut.
- Memiliki *setter* dan *getter* untuk mengakses setiap properti *private*.

4.4.6 Kelas Food

Kelas ini merepresentasikan suatu bahan makanan dalam suatu menu. Kelas ini memiliki properti-properti privat sebagai berikut:

- *id*, *name*, *alias*, *dataSource* dan *type* masing-masing merepresentasikan *id*, nama, alias atau nama lain, sumber data dan tipe bahan.
- *qty*. Properti ini merepresentasikan jumlah unit bahan dalam menu. Misalnya untuk gula 1 sendok, maka *qty* adalah 1.

- *activeUnitId*. Properti ini menyimpan *id* dari unit bahan yang saat ini aktif. Misalnya, gula memiliki 2 unit, sendok teh dengan *id* 0 dan sendok makan dengan *id* 1. Jika dalam menu dinyatakan "Gula 1 sendok teh", maka *activeUnitId* adalah 0.
- *isAdditional*. Properti ini menunjukkan apakah bahan merupakan bahan tambahan untuk sebuah menu.
- *weight*. Properti ini bertipe Tfn yang merepresentasikan berat bahan dalam sebuah menu. Misalnya, pada menu dinyatakan "Gula 1 sendok", sedangkan unit sedok faktor konversinya 15 gram. Maka *weight* adalah Tfn(0, 15, 0).
- *units*. Properti ini merupakan *array* yang berisi beberapa obyek kelas *Unit*.
- *nutrition*. Properti ini merupakan *array* yang berisi beberapa obyek kelas *Nutrition* yang berfungsi untuk menyimpan informasi nilai gizi bahan.

Kelas *Food* dapat diinisialisasi melalui sebuah *constructor* yang menerima sebuah obyek sebagai argumen. Setiap properti *private* kelas ini akan diset sesuai dengan properti dari obyek yang di-*passing* melalui argumen. Setiap properti *private* kelas ini memiliki *setter* dan *getter* yang dapat digunakan untuk akses dari fungsi lain.

4.4.7 Kelas StandardPortion

Kelas *StandardPortion* merepresentasikan suatu porsi penyajian standar untuk sebuah menu. Kelas ini memiliki dua properti *private*, yakni *label* dan *value*. *Label* merupakan representasi *string* dari *value*. Misalnya, untuk *value* 0.5 dapat diberi *label* "1/2".

Kelas *StandardPortion* memiliki satu *constructor* yang menerima obyek sebagai argumen. Nilai properti-properti *private* akan diset sesuai dengan properti dari obyek ini. Masing-masing properti *private* juga memiliki *setter* dan *getter* yang dapat digunakan fungsi-fungsi lain untuk mengakses properti ini.

4.4.8 Kelas Menu

Kelas ini merepresentasikan suatu menu. Kelas ini memiliki properti-properti *private* sebagai berikut:

- *id*, *name*, *category* dan *portion* masing-masing merepresentasikan *id*, nama, kategori dan porsi menu.
- *foods*. Merupakan *array* yang berisi beberapa objek kelas *Food* yang merepresentasikan tiap bahan makanan penyusun menu.
- *price*. Properti ini merupakan sebuah Tfn yang merepresentasikan harga menu.
- *activeStandardPortions*. Properti ini merupakan sebuah *array* yang menyimpan beberapa obyek kelas *StandardPortion*. Tiap obyek ini,

mewakili porsi penyajian standar yang dimiliki oleh menu. Misalnya menu "Soto Ayam" memiliki standar penyajian 1 porsi dan 2 porsi, maka *activeStandardPortions* adalah *array* berisi 2 buah obyek *StandardPortion*.

- *nutrition*. Properti ini merupakan *array* yang berisi beberapa obyek kelas *Nutrition* yang berfungsi untuk menyimpan informasi nilai gizi menu.

Kelas *Menu* memiliki sebuah *constructor* yang menerima sebuah obyek sebagai argumen. Setiap properti *private* kelas ini akan diset sesuai dengan properti dari obyek yang di-*passing* melalui argumen. Fungsi-fungsi yang dimiliki kelas ini yakni: *hasName*, *hasNutritionData* dan *hasActiveStandardPortions* yang masing-masing digunakan untuk mengecek kelengkapan data menu. Agar fungsi lain dapat mengakses properti-properti *private* yang dimiliki kelas ini maka untuk tiap properti *private* tersebut akan didefinisikan sebuah *setter* dan sebuah *getter*.

4.4.9 Kelas MenuSet

Kelas ini merepresentasikan sebuah *menu set*, yakni kumpulan dari beberapa menu. Kelas *MenuSet* memiliki spesifikasi sebagai berikut:

- Memiliki properti *private id, name* dan *menus*. *id* dan *name* masing-masing merepresentasikan *id* dan nama *menu set*. Sedangkan *menus* merupakan *array* yang berisi satu atau beberapa objek kelas *Menu*.
- Diinisialisasi dengan satu *constructor* yang menerima argumen berupa objek. Nilai properti-properti *private* kelas ini akan diset sesuai dengan properti dari objek tersebut.
- Memiliki fungsi *addMenu* yang digunakan untuk menambahkan objek *Menu* baru ke dalam *menus*.
- Memiliki fungsi *isDataComplete* yang dapat digunakan pada saat pengambilan data dari basis data untuk mengetahui apakah semua data yang diperlukan telah tersedia.
- Memiliki *setter* dan *getter* untuk mengakses setiap properti *private*.

4.5 Perancangan basis data

Perancangan basis data diperlukan untuk memodelkan struktur basis data sesuai dengan kebutuhan sistem. Perancangan meliputi penentuan tabel-tabel yang diperlukan, data-data yang akan dimuat dalam tabel tersebut serta hubungan atau *relationship* antar tabel.

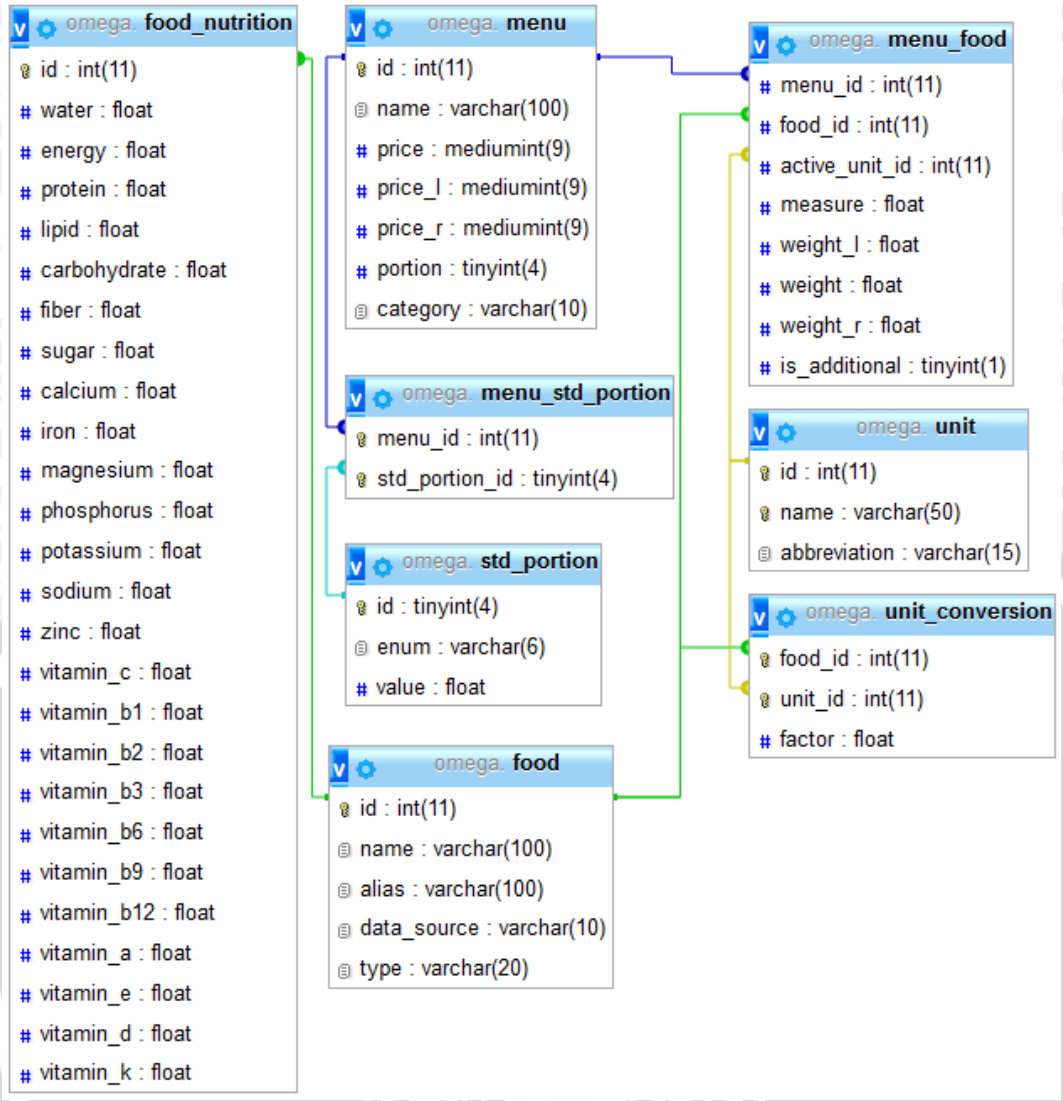
4.5.1 Perancangan basis data untuk data menu dan bahan makanan

Bagian ini akan membahas perancangan basis data yang berkaitan dengan data menu dan data bahan makanan. Untuk dapat menyimpan data-data tersebut, dibutuhkan struktur basis data dengan rancangan sebagai berikut:

- Data menu akan disimpan dalam tabel bernama *"menu"*.
- Data bahan makanan akan disimpan dalam tabel bernama *"food".*
- Data satuan bahan makanan disimpan dalam tabel bernama "unit".
- Data porsi penyajian standar disimpan dalam tabel bernama *"std_portion"*
- Data kandungan gizi bahan makanan disimpan dalam tabel bernama *"food_nutrition"*.
- Sebuah menu dapat terdiri dari satu atau banyak bahan dan sebuah bahan dapat dimiliki oleh satu atau banyak menu. Dengan demikian, hubungan yang terbentuk adalah *many-to-many relationship*. Untuk hubungan tipe ini, diperlukan satu lagi tabel tambahan yang disebut sebagai tabel *mapping*. Dalam hal ini yang berperan sebagai tabel *mapping* adalah tabel *"menu food"*. Dengan adanya tabel *mapping*, maka terbentuk hubungan sebagai berikut:
    - Satu menu dapat memiliki satu atau banyak *'menu food'*. Akan tetapi, satu buah *'menu food'* hanya dapat dimiliki oleh satu buah menu.
    - Satu *'menu food'* memiliki tepat satu bahan, tetapi suatu bahan dapat dimiliki oleh beberapa *'menu food'* yang berbeda.

  Tabel *'menu food'* juga memiliki beberapa *field* yang berfungsi untuk memberikan informasi tambahan pada *mapping* antara *'menu_id'* dan *'food_id'*. Misalnya, jika *'menu food'* yang memetakan menu_001 ke food_002 memiliki nilai *'is_additional'* bernilai 1, artinya food_002 pada menu_001 adalah sebagai bumbu atau bahan pelengkap.
- Sebuah bahan dapat memiliki satu atau beberapa *unit* dan sebuah *unit* dapat dimiliki oleh satu atau beberapa menu. Dengan demikian, hubungan yang terbentuk adalah *many-to-many relationship.* Untuk itu diperlukan sebuah tabel *mapping*, yakni tabel *"unit conversion"*. Tabel ini juga sekaligus menjelaskan berapa faktor konversi (ke dalam gram) dari sebuah *unit* untuk sebuah bahan tertentu.
- Sebuah menu dapat memiliki satu atau beberapa porsi penyajian standar dan satu porsi standar dapat dimiliki oleh beberapa menu yang berbeda. Dengan demikian terbentuk hubungan *many-to-many*. Hubungan ini memunculkan sebuah tabel *mapping* yang diberi nama *"menu_std_portion"*.
- Sebuah bahan memiliki tepat satu buah data kandungan gizi bahan sehingga membentuk sebuah hubungan *one-to-one relationship*.

Dari rancangan tersebut dapat dibangun sebuah basis data dengan struktur yang dapat digambarkan dalam bentuk *Physical Data Model* (PDM) pada Gambar 4.7.



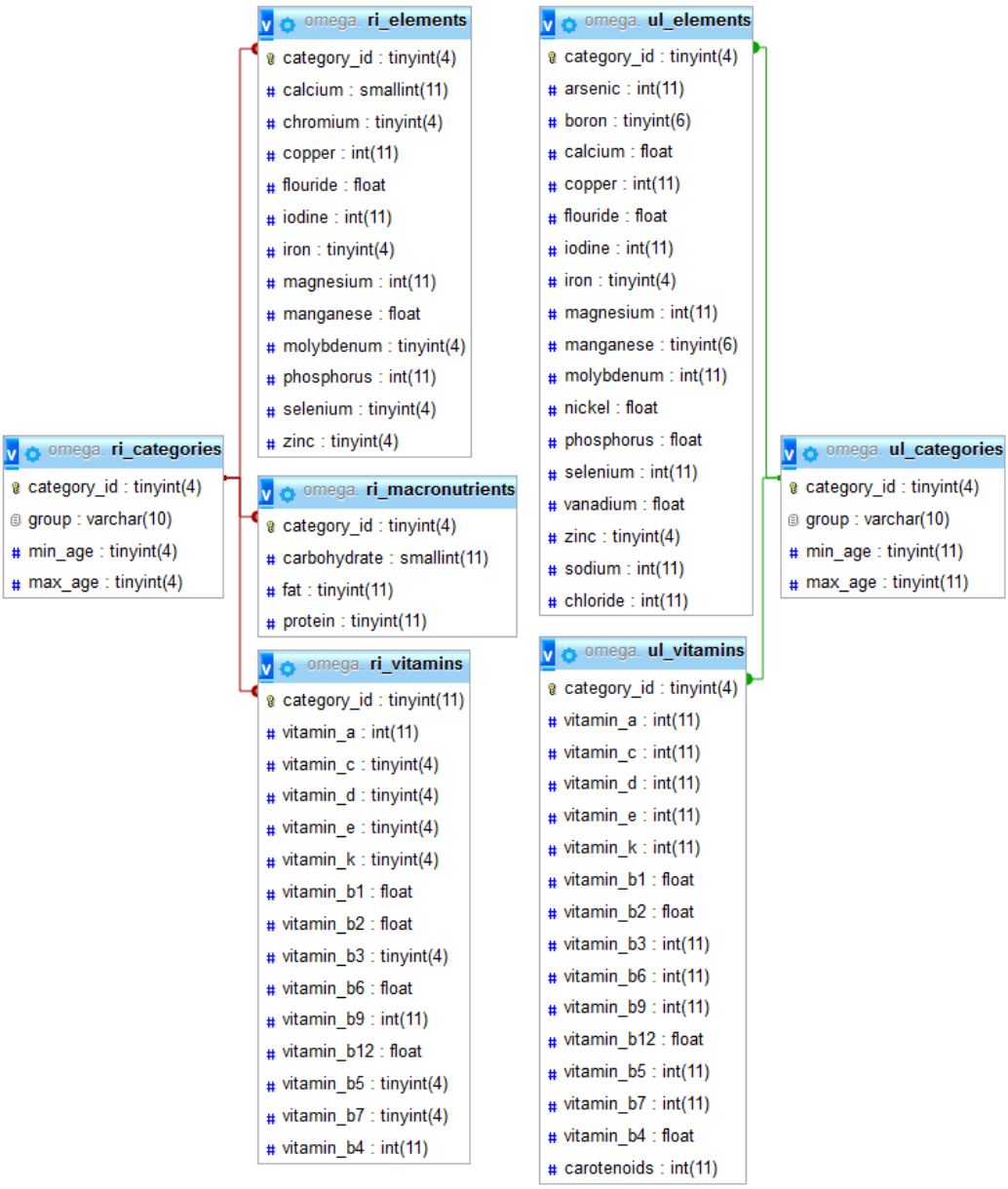
Gambar 4.7 Tabel-tabel yang berkenaan dengan menu dan bahan

4.5.2 Perancangan basis data untuk data *reference intakes* dan *upper level*

Bagian ini akan membahas perancangan basis data yang berkaitan dengan data *Reference Intakes* (RI) dan data *Upper Level* (UL). Kedua data ini strukturnya bisa dikatakan identik dan memiliki kriteria sebagai berikut:

- Data RI dan UL masing-masing memiliki data mengenai kategorisasi subjek berdasarkan kelompok *(group)* dan rentang usia. Data kategorisasi ini akan disimpan masing-masing pada tabel *"ri_categories"* dan *"ul_categories"*.
- Tiap data pada *"ul_categories"* atau *"ri_categories"* memiliki tepat satu nilai IR atau UL untuk nutrien tertentu *(one-to-one relationship)*. Nutrien-nutrien ini dikelompokkan menjadi 3 kelompok untuk IR *(macronutrients, vitamins* dan *elements)* dan 2 kelompok untuk UL *(vitamins* dan *elements)*.

- Tiap kelompok nutrien akan disimpan pada tabel berbeda. Sehingga diperlukan tabel: *"ri_macronutrients", "ri_vitamins", "ri_elements", "ul_vitamins"* dan *"ul_elements"*. Hubungan antara tabel tabel tersebut dapat digambarkan dalam bentuk *Physical Data Model* (PDM) pada Gambar 4.8.

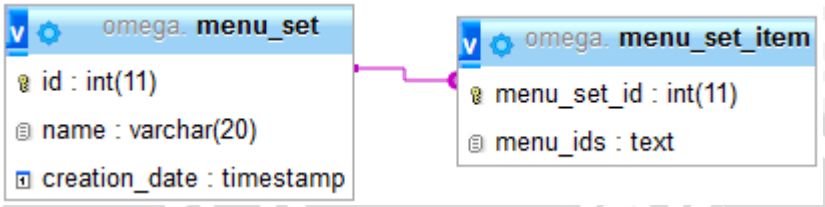


Gambar 4.8 Tabel-tabel yang berkenaan dengan RI dan UL

4.5.3 Perancangan basis data untuk data *menu set*

Untuk menyimpan data-data *menu set*, struktur basis data yang dibutuhkan cukup sederhana. Data *menu set* akan disimpan dalam tabel *"menu_set"*. Tiap *menu set* dapat memiliki satu atau lebih menu dan tiap menu dapat dimiliki oleh satu atau lebih *menu set*. Hubungan yang terbentuk adalah *one-to-one*

relationship dan memerlukan tabel *mapping*, yakni tabel *"menu_set_item"*. Sebenarnya, struktur yang biasa digunakan untuk kasus seperti ini adalah *'menu_id'* pada tabel *"menu_set_item"* adalah *foreign key* yang merujuk pada tabel menu. Namun, penulis memilih struktur berbeda, di mana *id* menu dalam *"menu_set_item"* disimpan dalam bentuk teks dan tidak merujuk ke tabel manapun. Struktur ini memberi kelebihan yakni sebuah *menu set* dapat memiliki dua atau lebih menu yang sama. Struktur tabel untuk menyimpan data *menu set* dapat digambarkan dalam bentuk *Physical Data Model* (PDM) pada Gambar 4.9.



Gambar 4.9 Tabel-tabel yang berkenaan dengan *menu set*

4.5.4 Stored procedure

Untuk mempermudah proses pengambilan data sekaligus mengurangi lalu lintas data antara sisi klien *(client side)* dan sisi *server (server side)*, diperlukan beberapa *stored procedure*. *Stored procedure* adalah satu atau beberapa *SQL statement* untuk melakukan tugas tertentu yang dienkapsulasi dan dieksekusi di *server*. Sebagian besar *stored procedure* yang diperlukan berkaitan dengan tugas-tugas *Create, Read, Update, Delete* (CRUD) dan tidak terlalu relevan dengan fokus penelitian. Untuk itu penulis membatasi bahasan hanya pada beberapa *stored procedure* yang dirasa relevan.

- menu.nutrition.total

  Digunakan untuk mendapatkan total nutrisi untuk tiap-tiap nutrien dari sebuah menu. Menerima parameter berupa *id* menu dan mengembalikan data total nutrisi menu dalam bentuk *fuzzy*.

- nutrient.getRi

  Digunakan untuk mendapatkan data RI berdasarkan data profil subjek. Menerima parameter berupa kelompok subjek ('anak-anak', 'laki-laki' atau 'perempuan') dan usia subjek. *Stored procedure* ini mengembalikan data RI dari subjek yang bersangkutan.

- nutrient.getUl

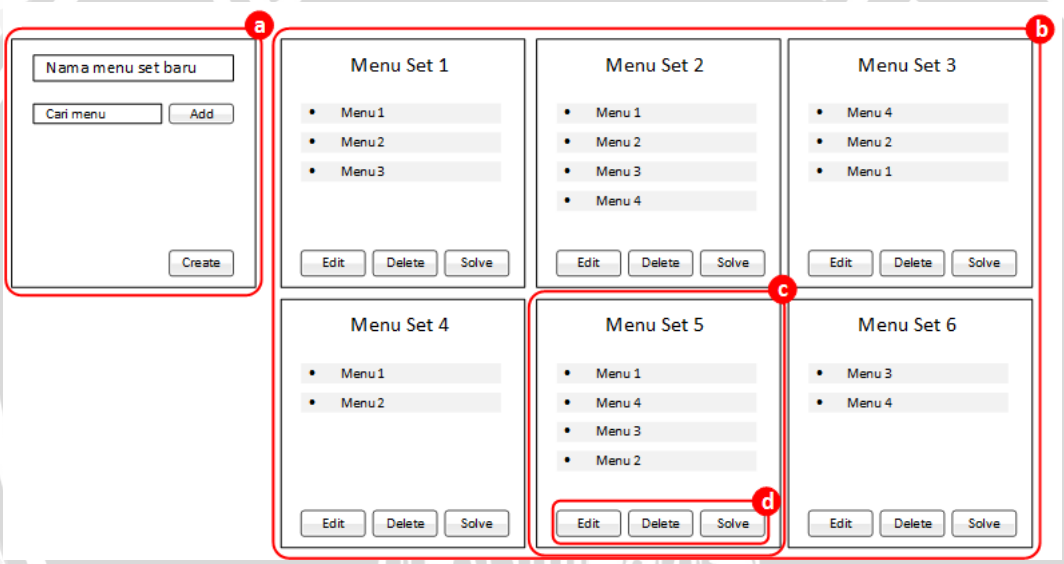
  Identik dengan nutrient.getRi, *stored procedure* ini digunakan untuk mendapatkan data UL berdasarkan data profil subjek. *Stored procedure* ini menerima parameter berupa kelompok subjek ('anak-anak', 'laki-laki' atau 'perempuan') dan usia subjek. Data kembalian dari *stored procedure* ini adalah data UL dari subyek bersangkutan.

4.6 Perancangan antar muka pengguna

Perancangan antarmuka diperlukan untuk memodelkan antarmuka aplikasi dengan pengguna. Perancangan ini bertujuan untuk menghasilkan aplikasi yang mudah digunakan serta memiliki *layout* (penataan) yang efisien namun tetap informatif. Pada penelitian ini, antarmuka pengguna akan dibagi menjadi antarmuka *"Menu Set Manager"*, antarmuka *"Solver"* dan antarmuka untuk pengujian.

4.6.1 Antarmuka *menu set manager*

Antarmuka ini dapat digunakan untuk memilih menu-menu yang akan dioptimasi. Menu-menu yang telah dipilih dapat disimpan dan di-*edit*. Tiap menu set yang ada pada antarmuka ini dapat 'dikirim' ke antarmuka *"Solver"* untuk dicari kombinasi optimalnya. Antarmuka ini memiliki rancangan seperti pada Gambar 4.10.



Gambar 4.10 Rancangan antarmuka *menu set manager*

Bagian-bagian dari antarmuka *"Menu Set Manager"* dapat dijelaskan sebagai berikut:

a) Panel *Menu Set* Baru, digunakan untuk membuat *menu set* baru. Memiliki *field* untuk memberikan nama *menu set* serta menambahkan menu-menu yang ingin dimasukkan ke dalam *menu set*.

b) Daftar *Menu Set*, merupakan daftar dari semua *menu set* yang disimpan dalam basis data.

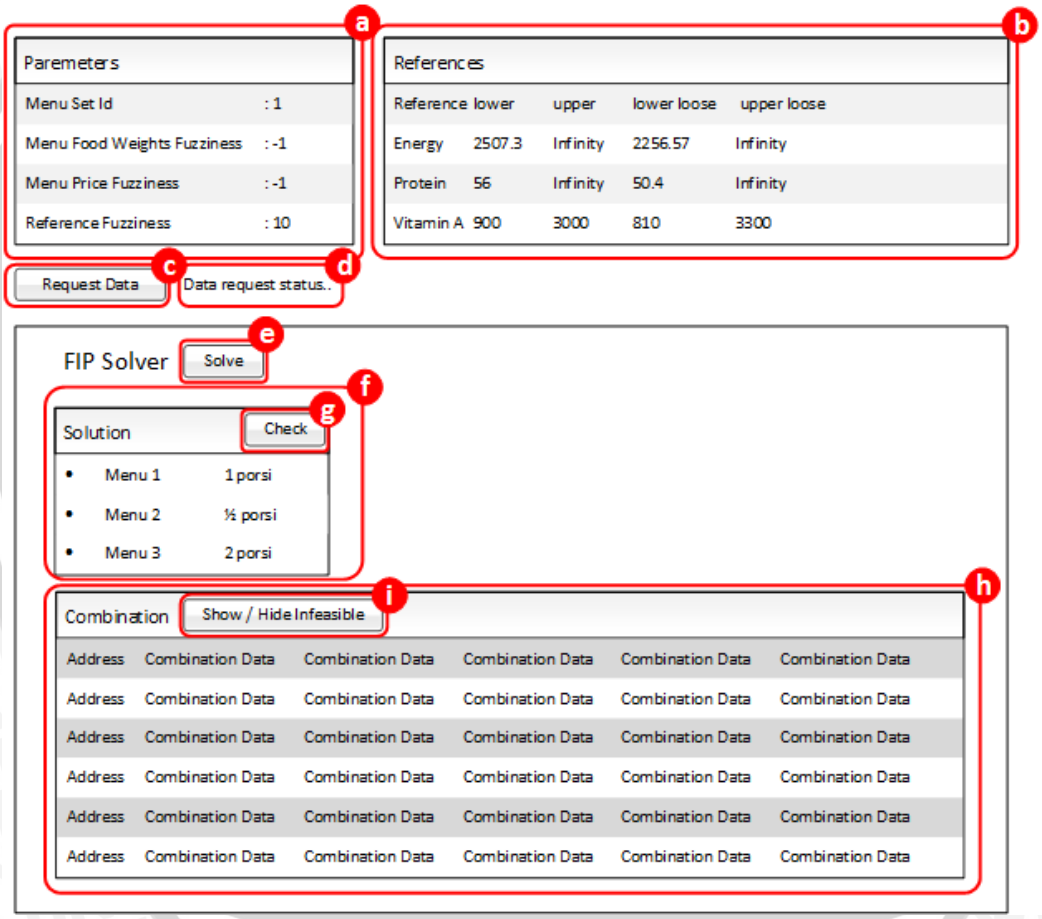
c) *Menu Set*, menampilkan data *menu set*. Memiliki label nama, daftar menu dan tombol aksi.

d) Tombol Aksi *Menu Set*. Digunakan untuk melakukan aksi tertentu pada *menu set*, yakni:

- *Edit*. Mengubah nama dan/atau isi *menu set*.
- *Delete*. Menghapus *menu set* dari basis data.
- *Solve*. Memecahkan permasalahan yang didefinisikan oleh *menu set*, yakni mencari kombinasi porsi paling optimal dari menu-menunya. Pemecahan ini akan dilakukan di antarmuka *"Solver"*.

4.6.2 Antarmuka *solver*

Antarmuka ini digunakan untuk mencari kombinasi optimal dari menu-menu yang ada dalam *menu set* terpilih. Pada antarmuka ini juga dapat dilakukan *exhaustive check*, yaitu membandingkan solusi yang diperoleh dari FIP dengan seluruh kemungkinan solusi. Parameter-parameter *solver* diatur melalui URL. Antarmuka *"Solver"* memiliki bagian-bagian utama seperti pada Gambar 4.11.



Gambar 4.11 Rancangan antarmuka *solver*

Bagian-bagian antarmuka *solver* dapat dijelaskan sebagai berikut:

a) Panel Parameter, menampilkan parameter-parameter yang digunakan oleh *solver*.

b) Panel *References*, menampilkan batasan-batasan yang diterapkan pada permasalahan yang akan diselesaikan.

c) Tombol *Request Data*, menginstruksikan aplikasi untuk mengirimkan *request* beserta parameter-parameternya ke *server* basis data.

d) Status Data, menampilkan status data yang sedang di-*request*.

e) Tombol *Solve,* menginstruksikan aplikasi untuk memecahkan permasalahan.

f) Panel Solusi, menampilkan solusi dari permasalahan. Jika permasalahan tidak memiliki solusi, panel ini akan menampilkan *'no feasible solution'*.

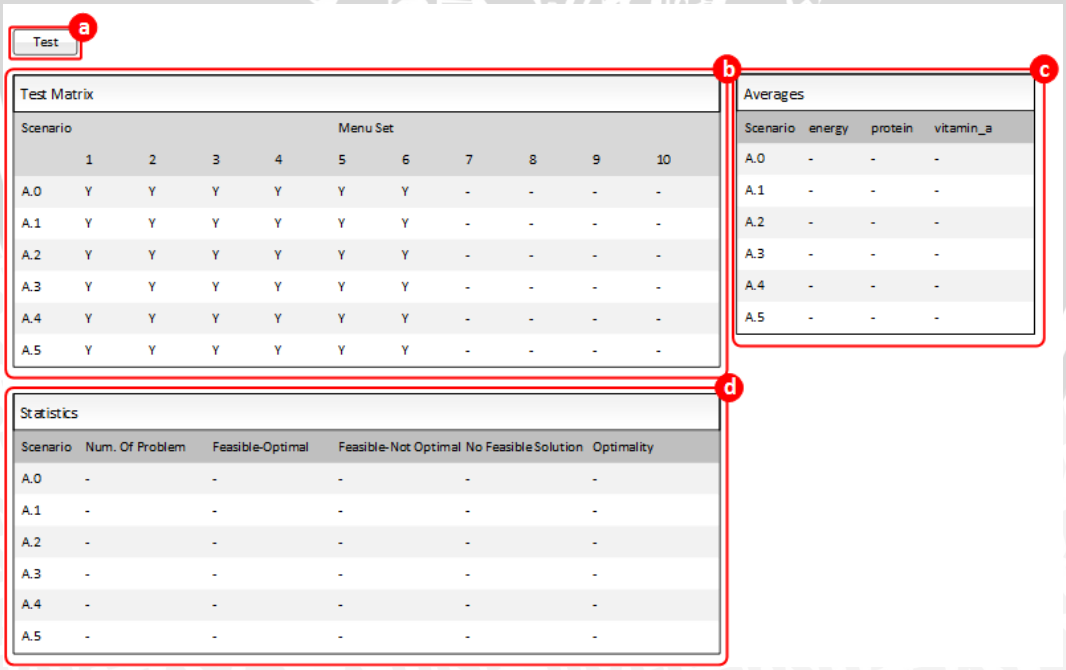
g) Tombol *Check*, menginstruksikan aplikasi untuk membangkitkan seluruh kemungkinan solusi. Hasil pembangkitan ini dapat digunakan untuk menilai optimalitas algoritma FIP.

h) Panel Kombinasi, menampilkan semua kombinasi hasil pembangkitan.

i) Tombol *Show/Hide Infeasible*, digunakan untuk menampilkan atau menyembunyikan kombinasi-kombinasi hasil pembangkitan yang tidak memenuhi.

4.6.3 Antarmuka pengujian

Antarmuka ini diperlukan dalam tahap pengujian algoritma dan analisa hasil. Antarmuka ini akan menampilkan matriks hasil pengujian, statistik pengujian, serta rata-rata capaian untuk skenario-skenario yang didefinisikan. Antarmuka pengujian memiliki rancangan seperti pada Gambar 4.12.



Gambar 4.12 Rancangan antarmuka *lab*

Bagian-bagian dari antarmuka pengujian dapat dijelaskan sebagai berikut:

a) Tombol *Test*, digunakan untuk memulai proses pengujian.

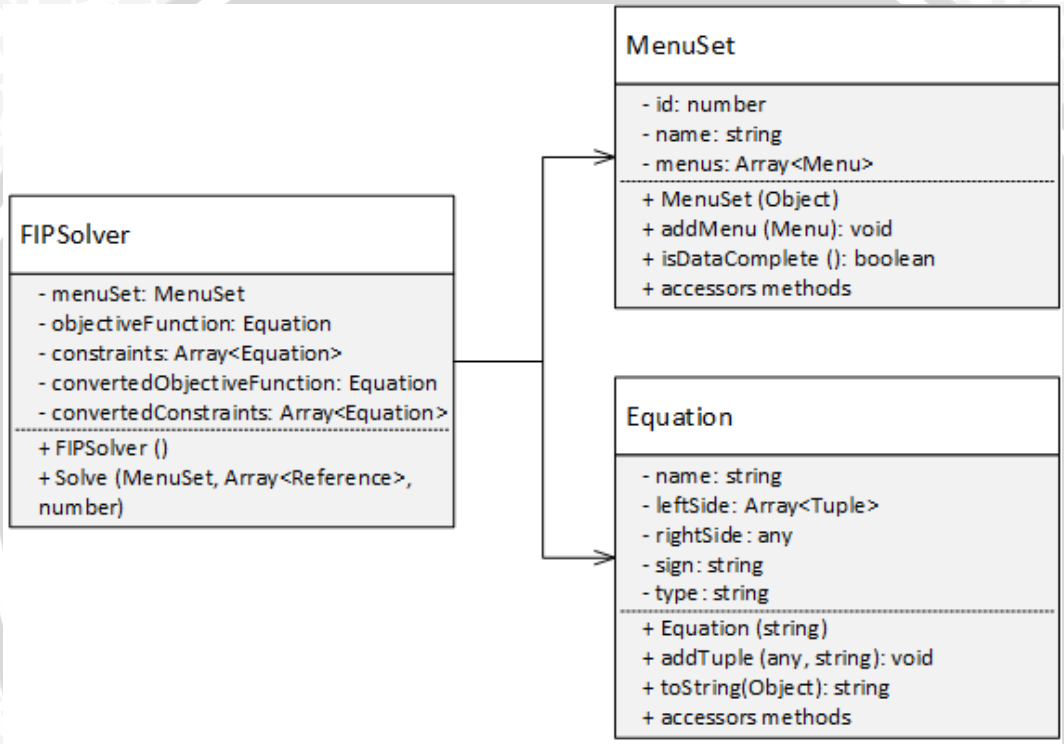
b) Matriks Hasil Pengujian, menampilkan hasil pengujian tiap *menu set* dari tiap-tiap skenario.

c) Tabel Rata-Rata Capaian, menampilkan rata-rata capaian untuk tiap skenario.

d) Tabel Statistik Pengujian, menampilkan statistik pengujian.

4.7 Implementasi algoritma

Algoritma *Fuzzy Integer Programming* diimplementasikan ke dalam kelas tersendiri, yakni kelas *"FIPSolver"*. Arsitektur dari kelas FIPSolver dapat digambarkan dalam bentuk *class diagram* pada Gambar 4.13.



Gambar 4.13 *Class diagram* FIPSolver

Kelas ini memiliki properti *private* sebagai berikut:

- *menuSet*. Properti bertipe *MenuSet* ini mewakili permasalahan yang akan dioptimasi. *menuSet* berisi beberapa menu berupa objek kelas *Menu*. Tiap menu memiliki data mengenai harga dan porsi penyajian standar yang diambil dari basis data. Tiap menu juga memiliki data kandungan gizi (berupa *array* berisi objek-objek kelas *Nutrition*) yang didapatkan dari perhitungan yang dilakukan akan oleh *server* basis data.
- *objectiveFunction*. Properti ini bertipe *Equation* dan berfungsi untuk menyimpan persamaan fungsi tujuan dari permasalahan yang akan diselesaikan.

- *constraints*. Properti ini berupa *array* yang nantinya akan berisi beberapa objek dari kelas *Equation*. Properti ini menyimpan persamaan-persamaan batasan dari permasalahan yang akan diselesaikan.
- *convertedObjectiveFunction*. Properti ini bertipe *Equation* dan akan menyimpan persamaan fungsi tujuan yang telah dikonversi.
- *convertedConstraints*. Properti ini berupa *array* yang nantinya akan menampung persamaan-persamaan batasan hasil konversi berupa beberapa objek kelas *Equation*.

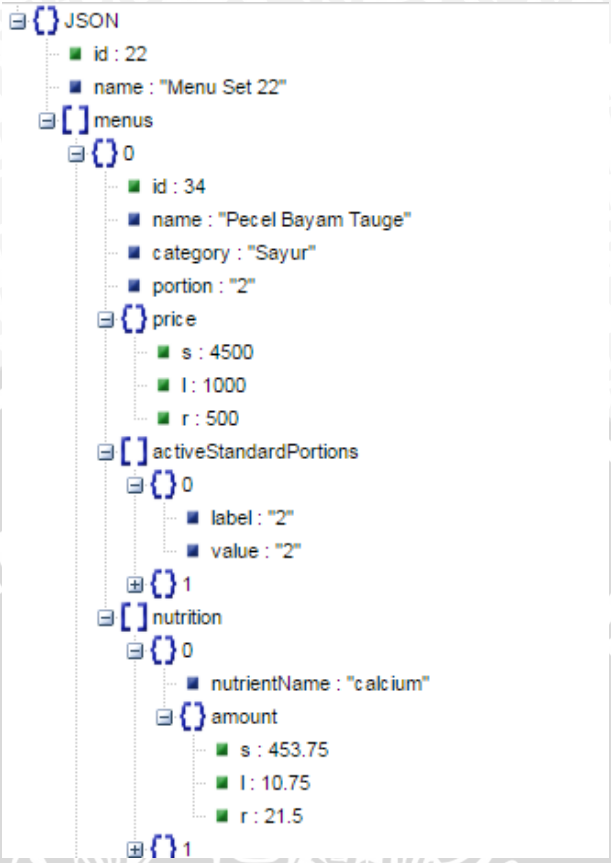
Constructor FIPSolver tidak melakukan apapun selain melakukan inisialisasi properti *constraints* dan *convertedConstraints* sebagai *array* kosong. Data-data yang dibutuhkan oleh kelas ini akan dijadikan argumen bagi fungsi utama, yakni fungsi *solve*.

4.7.1 Fungsi *solve*

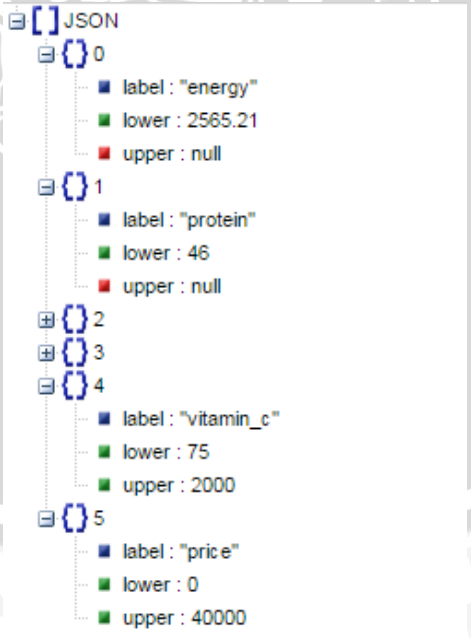
Fungsi *solve* adalah fungsi utama dari kelas FIPSolver. Setelah kelas FIPSolver diinisiasi dengan memanggil *constructor*-nya, algoritma *Fuzzy Integer Programming* dapat dijalankan dengan memanggil fungsi *solve*. Argumen yang dibutuhkan fungsi *solve* adalah:

- *menuSet*. Data dengan tipe *MenuSet* yang ingin dicari solusi optimalnya. Gambar 4.14 adalah contoh data *menu set* dalam format JSON *(JavaScript Object Notation)*.
- *references*. Merupakan *array* yang berisi beberapa objek *Reference* yang merepresentasikan batasan-batasan nilai nutrien. Mengingat batasan harga juga dapat direpresentasikan dengan sebuah *Reference*, maka batasan harga juga dapat dimasukkan ke dalam *array references* dengan label *'price'*. Nantinya, *Reference* dengan label ini akan diperlakukan secara khusus. Contoh data *references* dapat dilihat pada Gambar 4.15.
- *referenceTolerance*. Merupakan angka (dalam persen) yang menyatakan seberapa jauh nilai nutrien boleh menyimpang dari batas yang ditentukan oleh *reference*. Misalnya, *reference* untuk "protein" memiliki batas atas 50, jika *refTolerance* diberi nilai 10 (artinya 10%) maka batas atas sekarang menjadi 55. Jika *reference* tersebut juga memiliki batas bawah, misalnya 20, maka batas bawah sekarang menjadi 18. Argumen ini tidak berlaku pada *Reference* dengan label khusus, yakni *'price'*, karena untuk penelitian ini batasan harga nilainya dianggap pasti (tidak buram).

repository.ub.ac.id



Gambar 4.14 Contoh data *menu set*



Gambar 4.15 Contoh data *references*

UNIVERSITAS BRAWIJAYA

Ketika dipanggil, fungsi ini akan membangun persamaan-persamaan yang dibutuhkan sesuai dengan data *menu set* dan *reference-reference* yang dimasukkan melalui argumen. Persamaan-persamaan tersebut kemudian akan dikonversi untuk dapat diselesaikan dengan menggunakan *Integer Programming*. Setelah model matematika dari permasalahan telah siap, fungsi akan memanggil *library Integer Programming Solver* untuk mencari solusi dari model tersebut. Berikut adalah *pseudo code* dari fungsi *solve* kelas *FIPSolver*:

```
solve (menuSet: MenuSet, refs: Array of Reference, refTolerance: number): void
   objectiveFunction <-- buildObjectiveFunction(menuSet)

   constraints <-- Array
   lowerRef <-- semua reference dalam refs dengan properti isLowerImportant=true
   upperRef <-- semua reference dalam refs dengan properti isUpperImportant=true
   priceRef <-- reference dalam refs dengan properti isPriceReference=true
   lowerConstraints <--buildConstraintsFromReference (menuSet,
                       lowerRef, 'lower', refTolerance)
   upperConstraints <--buildConstraintsFromReference (menuSet,
                       upperRef, 'upper', refTolerance)
   priceConstraints <--buildConstraintsFromReference (menuSet,
                       priceRef, 'price', 0)
   additionalConstraints <-- buildAdditionalConstraints(menuSet)
   tambahkan tiap constraint dalam lowerConstraints ke dalam constraints
   tambahkan tiap constraint dalam upperConstraints ke dalam constraints
   tambahkan tiap constraint dalam priceConstraints ke dalam constraints
   tambahkan tiap constraint dalam additionalConstraints ke dalam constraints

   objectiveFunction <-- convertObjectiveFunction(objectiveFunction)
   constraints <-- convertConstraints(constraints)

   ubah objectiveFunction dan constraint-constraint, ke dalam bentuk String
   selesaikan dengan Integer Programming Solver
```

4.7.2 Fungsi *buildObjectiveFunction*

Fungsi *buildObjectiveFunction* berfungsi untuk membangun persamaan fungsi tujuan dari permasalahan yang ingin diselesaikan. Fungsi ini menerima satu argumen, yakni sebuah *menu set*, dan mengembalikan persamaan fungsi tujuan berupa sebuah objek *Equation*.

Pada awal fungsi, Fungsi akan membuat sebuah objek *Equation* baru dengan ruas kanan diberi nilai 'Z' dan tanda persamaan berupa "=". Selanjutnya, fungsi akan mulai menyusun ruas kiri persamaan berdasarkan data pada menu set. Setelah ruas kiri tersusun, fungsi akan mengembalikan objek *Equation* tersebut sebagai nilai kembalian. Berikut adalah *pseudo code* dari fungsi *buildObjectiveFunction*:

```
buildObjectiveFunction (menuSet: MenuSet): Equation
   objectiveFunction <-- Equation, dengan rightSide='Z', sign='='
   foreach menu in menuSet, menuIndex as index
      foreach standardPortion of menu, portionIndex as index
         coeff <-- Tfn, hasil dari menu.price dikalikan dengan standardPortion
         var <-- String, 'M' + menuIndex + portionIndex
         tambahkan pasangan coeff dan var ke bagian leftSide objectiveFunction
      end foreach
   end foreach
   return objectiveFunction
```

4.7.3 Fungsi *buildConstraintsFromReference*

Fungsi *buildConstraintsFromReference* berfungsi untuk membangun persamaan-persamaan batasan dari permasalahan yang ingin diselesaikan berdasarkan batasan-batasan yang didefinisikan melalui *references*. Fungsi ini menerima empat buah argumen, yaitu:

- *menuSet*. *Menu set* yang ingin dicari solusi optimalnya.
- *references*. Sebuah *array* yang berisi beberapa objek kelas *Reference* yang merepresentasikan batasan-batasan nilai nutrien atau harga.
- *constraintType*. *"lower"* jika yang ingin dibangun adalah batas-batas bawah, *"upper"* jika yang ingin dibangun adalah batas-batas atas. *"price"* jika yang ingin dibangun adalah persamaan batasan harga.
- *referenceTolerance*. Merupakan angka (dalam persen) yang menyatakan seberapa jauh nilai nutrien boleh menyimpang dari batas yang ditentukan oleh *reference*. Nilai *referenceTolerance* tidak berlaku untuk batasan harga karena untuk penelitian ini batasan harga dianggap tidak buram.

Saat dipanggil, fungsi ini akan membangun batasan untuk setiap *reference* dalam *references*. Setiap batasan yang dibangun oleh fungsi ini memiliki tipe persamaan sesuai dengan argumen *constraintType* (*"lower"*, *"upper"*, atau *"price"*). Argumen *constraintType* juga menentukan nilai dari ruas kanan dan tanda persamaan batasan. Setelah tipe, ruas kanan dan tanda persamaan batasan ditentukan, fungsi akan mulai membangun ruas kiri persamaan. Ruas kiri dibangun dengan menambahkan pasangan-pasangan koefisien dan variabel ke bagian *leftSide* dari persamaan batasan yang sedang dibangun. Berikut adalah *pseudo code* dari fungsi *buildConstraintsFromReference*:

```
buildConstraintsFromReference     (menuSet: MenuSet, refs: Array of Reference,
                                  constraintType: String, refTolerance: number
                                  ): Array of Equation
   constraints <-- Array
   foreach reference in refs
      constraint <-- Equation, dengan type=constraintType
      switch (constraintType)
```

```
        case 'lower'
           constraint.sign <-- '>='
           lowerRef <-- reference.lower
           constraint.rightSide <-- Tfn, dengan   s=lowerRef,
                                                  l=refTolerance/100*lowerRef,
                                                  r=refTolerance/100*lowerRef
        case 'upper'
           constraint.sign <-- '<='
           upperRef <-- reference.upper
           constraint.rightSide <-- Tfn, dengan   s=upperRef,
                                                  l=refTolerance/100*upperRef,
                                                  r=refTolerance/100*upperRef
        case 'price'
           constraint.sign <-- '<='
           upperRef <-- reference.upper
           constraint.rightSide <-- Tfn, dengan   s=upperRef,
                                                  l=0,
                                                  r=0
     foreach menu in menuSet, menuIndex as index
        foreach standardPortion of menu, portionIndex as index
           if (constraintType is 'price')
              amount <-- Tfn, menu.price
           else
              amount <-- Tfn, nilai menu.nutrition.amount di mana
                         menu.nutrition.nutrientName = reference.label
           coeff <-- Tfn, amount dikalikan dengan standardPortion
           var <-- 'M' + menuIndex + portionIndex
           tambahkan pasangan coeff dan var ke bagian leftSide constraint
        end foreach
     end foreach
     tambahkan constraint ke dalam constraints
  end foreach
  return constraints
```

4.7.4 Fungsi *buildAdditionalConstraints*

Fungsi ini digunakan untuk membangun batasan-batasan tambahan yang digunakan untuk memastikan bahwa tiap menu yang ada dalam *menu set* disajikan tepat satu kali. Fungsi *buildAdditionalConstraints* menerima satu argumen, yakni *menu set* yang ingin dioptimasi, dan mengembalikan *array* yang berisi persamaan-persamaan batasan tambahan yang telah dibangun. Fungsi akan membangun persamaan batasan tambahan sebanyak jumlah menu dalam *menu set*. Tiap persamaan batasan memiliki nilai ruas kanan 1, tanda persamaan berupa "=" dan tipe persamaan *"additional"*. Ruas kiri dibangun dengan menambahkan pasangan-pasangan koefisien dan variabel pada bagian *leftSide* persamaan batasan. Setiap koefisien ruas kiri diberi nilai 1. Berikut adalah *pseudo code* fungsi *buildAdditionalConstraints*:

```
buildAdditionalConstraints (menuSet: MenuSet) : Array of Equation
   additionalConstraints <-- Array
   foreach menu in menuSet, menuIndex as index
      additionalConstraint <-- Equation, dengan    rightSide=1, sign='=',
                                                   type='additional'
      foreach standardPortion of menu, portionIndex as index
         coeff <-- 1
         var <-- 'M' + menuIndex + portionIndex
         tambahkan pasangan coeff dan var ke bagian leftSide additionalConstraint
      end foreach
      tambahkan additionalConstraint ke dalam additionalConstraints
   end foreach
   return additionalConstraints
```

4.7.5 Fungsi *convertObjectiveFunction*

Fungsi *convertObjectiveFunction* digunakan untuk mengonversi persamaan fungsi tujuan yang masih melibatkan bilangan *fuzzy* menjadi persamaan fungsi tujuan biasa. Fungsi ini menerima argumen berupa persamaan fungsi tujuan yang akan dikonversi dan mengembalikan persamaan hasil konversi. Konversi dilakukan berdasarkan aturan konversi FIPP ke MOIPP (Dash & Dash, 2012). Berikut adalah *pseudo code* dari fungsi *convertObjectiveFunction*:

```
convertObjectiveFunction (objectiveFunction: Equation): Equation
   foreach pasangan coeff dan var pada bagian leftSide objectiveFunction
      coeff <-- coeff.s + coeff.l + coeff.r
   end foreach
   return objectiveFunction
```

4.7.6 Fungsi *convertConstraints*

Fungsi *convertConstraints* digunakan untuk mengonversi persamaan-persamaan batasan yang melibatkan bilangan *fuzzy* menjadi persamaan-persamaan batasan biasa. Fungsi ini menerima argumen berupa *array* yang berisi persamaan-persamaan yang akan dikonversi dan mengembalikan *array* berisi persamaan-persamaan hasil konversi. Konversi dilakukan berdasarkan aturan konversi FIPP ke MOIPP (Dash & Dash, 2012). Berikut adalah *pseudo code* dari fungsi *convertConstraints*:

```
convertConstraints (constraints: Array of Equation) : Array of Equation
   convertedConstraints <-- Array
   // Rule-1
   foreach constraint in constraints, kecuali constraint-constraint 'additional'
      foreach pasangan coeff dan var pada bagian leftSide constraint
         coeff <-- coeff.s
      end foreach
      constraint.rightSide <-- constraint.rightSide.s
      tambahkan constraint ke convertedConstraints
   end foreach
```

```
// Rule-2
foreach constraint in constraints, kecuali constraint-constraint 'additional'
   foreach pasangan coeff dan var pada bagian leftSide constraint
      coeff <-- coeff.s - coeff.l
   end foreach
   constraint.rightSide <-- constraint.rightSide.s - constraint.rightSide.l
   tambahkan constraint ke convertedConstraints
end foreach

// Rule-3
foreach constraint in constraints, kecuali constraint-constraint 'additional'
   foreach pasangan coeff dan var pada bagian leftSide constraint
      coeff <-- coeff.s + coeff.r
   end foreach
   constraint.rightSide <-- constraint.rightSide.s + constraint.rightSide.r
   tambahkan constraint ke convertedConstraints
end foreach

tambahkan tiap constraint 'additional' dalam constraints ke dalam
convertedConstraints
return convertedConstraints
```

4.7.7 Integer programming solver

Untuk menyelesaikan permasalahan *Integer Programming* hasil konversi, digunakan sebuah *JavaScript library* bernama "Glpk.js". Glpk.js merupakan singkatan dari *GNU Linear Programming Kit for JavaScript* (Gourvest, 2016). *Library* ini menerima masukan berupa model permasalahan dengan format tertentu dan mengembalikan objek yang merupakan solusi dari masalah tersebut.

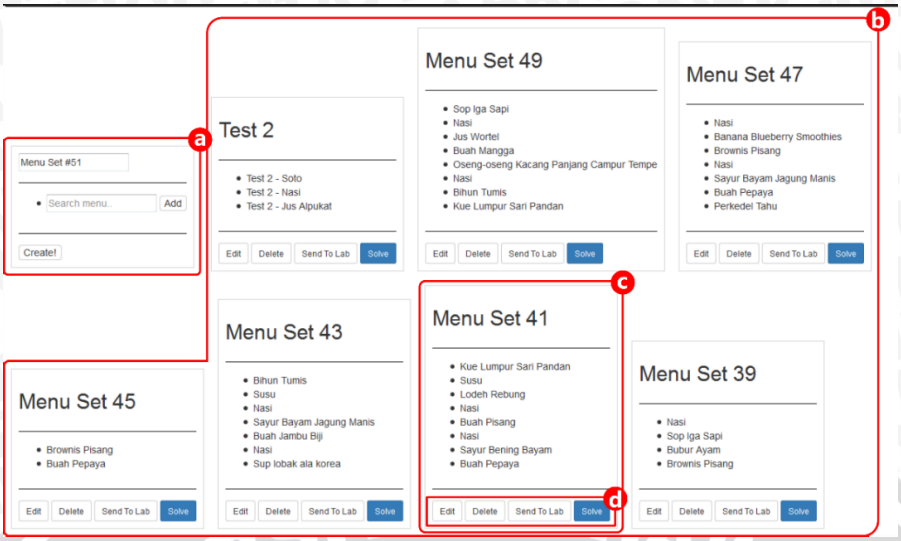
4.8 Implementasi antar muka pengguna

Setelah algoritma berhasil diimplementasikan, selanjutnya dilakukan implementasi antarmuka pengguna. Implementasi algoritma kemudian diintegrasikan dengan antarmuka ini sehingga menjadi sebuah sistem yang siap digunakan. Sesuai dengan rancangan, antarmuka terbagi menjadi 3, yakni antarmuka "*Menu Set Manager*", antarmuka "*Solver*", dan antarmuka pengujian.

4.8.1 Implementasi antarmuka *menu set manager*

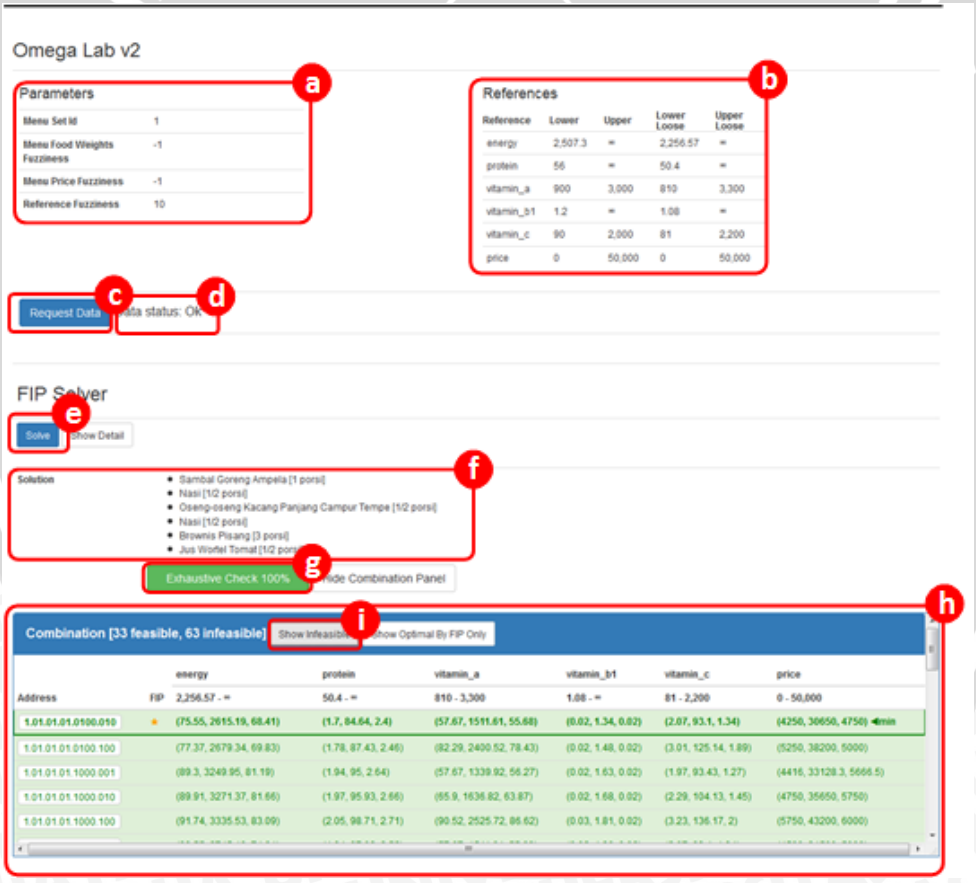
Hasil implementasi dari rancangan antarmuka *menu set manager* adalah seperti pada Gambar 4.16. Sesuai dengan rancangan, bagian-bagian antarmuka ini adalah:

a) Panel *Menu Set* Baru
b) Daftar *Menu Set*
c) *Menu Set*
d) Tombol Aksi *Menu Set*.



Gambar 4.16 Hasil implementasi antarmuka *menu set manager*

4.8.2 Implementasi antarmuka *solver*



Gambar 4.17 Hasil implementasi antarmuka *solver*

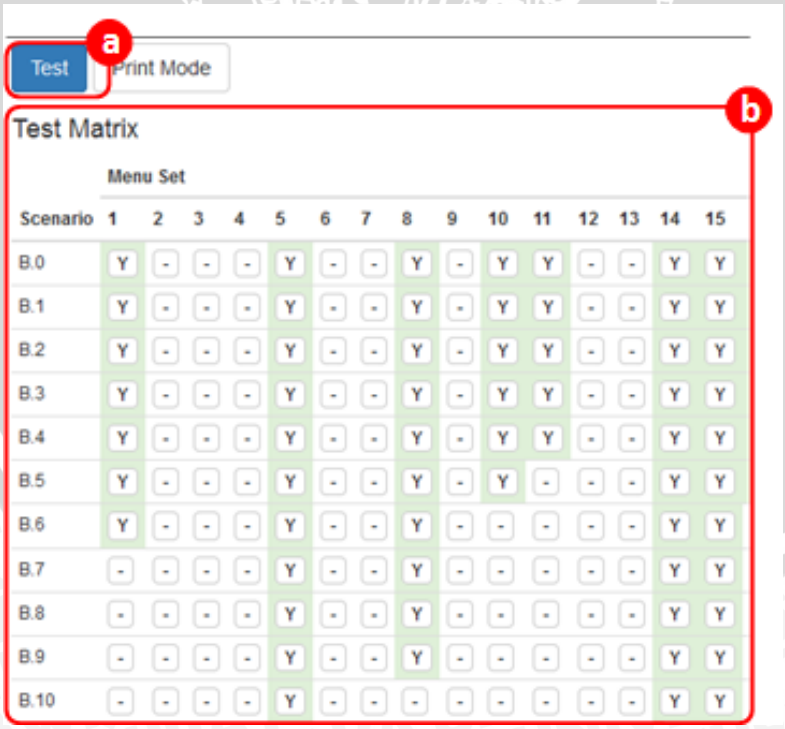
Hasil implementasi dari rancangan antarmuka *solver* adalah seperti pada Gambar 4.17. Sesuai dengan rancangan, antarmuka ini memiliki bagian-bagian utama sebagai berikut:

a) Panel Parameter
b) Panel *References*
c) Tombol *Request Data*,
d) Status Data
e) Tombol *Solve*
f) Panel Solusi
g) Tombol *Check*
h) Panel Kombinasi
i) Tombol *Show/Hide Infeasible*.

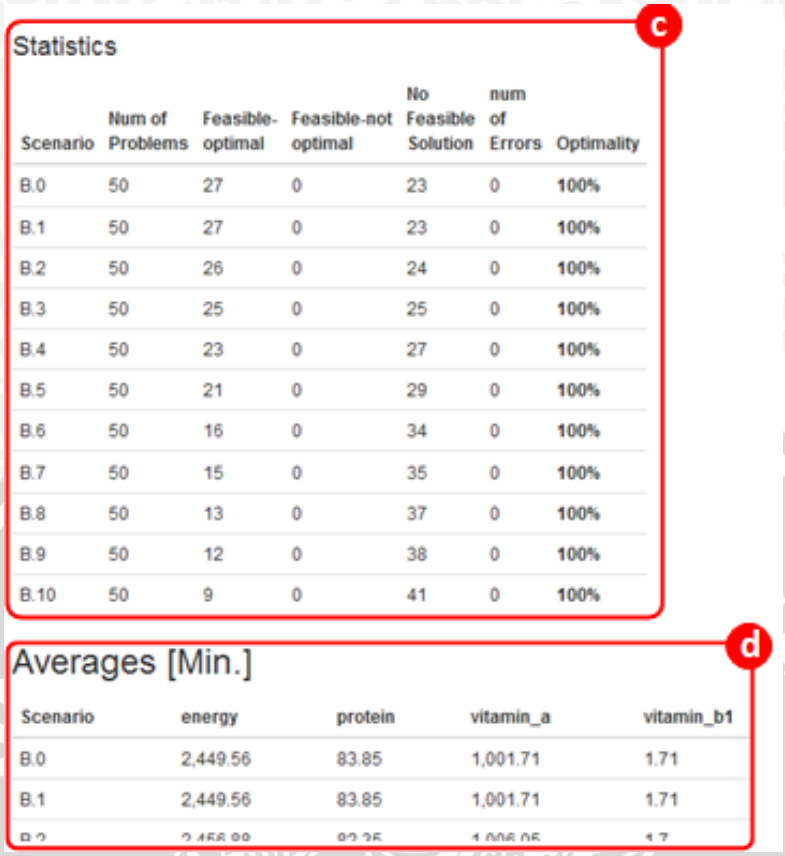
4.8.3 Implementasi antarmuka pengujian

Hasil implementasi dari rancangan antarmuka *solver* adalah seperti pada Gambar 4.18 dan Gambar 4.19. Sesuai dengan rancangan, bagian-bagian antarmuka ini adalah:

a) Tombol *Test*
b) Matriks Hasil Pengujian
c) Tabel Statistik Pengujian
d) Tabel Rata-Rata Capaian.



Gambar 4.18 Hasil implementasi antarmuka pengujian (1)



Statistics

Scenario	Num of Problems	Feasible-optimal	Feasible-not optimal	No Feasible Solution	num of Errors	Optimality
B.0	50	27	0	23	0	100%
B.1	50	27	0	23	0	100%
B.2	50	26	0	24	0	100%
B.3	50	25	0	25	0	100%
B.4	50	23	0	27	0	100%
B.5	50	21	0	29	0	100%
B.6	50	16	0	34	0	100%
B.7	50	15	0	35	0	100%
B.8	50	13	0	37	0	100%
B.9	50	12	0	38	0	100%
B.10	50	9	0	41	0	100%

Averages [Min.]

Scenario	energy	protein	vitamin_a	vitamin_b1
B.0	2,449.56	83.85	1,001.71	1.71
B.1	2,449.56	83.85	1,001.71	1.71

Gambar 4.19 Hasil implementasi antarmuka pengujian (2)

BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil dari pengujian yang telah dilakukan berdasarkan strategi-strategi pengujian yang telah dibahas pada bab "Metodologi". Bab ini akan dibagi menjadi dua subbab. Pada subbab pertama akan disajikan data-data hasil pengujian, sedangkan pada subbab ke-dua akan dibahas mengenai apa saja yang dapat ditemukan atau disimpulkan dari data hasil pengujian tersebut.

5.1 Hasil pengujian

Pada subbab ini akan akan disampaikan data-data hasil pengujian dari "Pengujian A" hingga "Pengujian D". Untuk mempermudah, skenario-skenario pengujian yang sebelumnya telah dibahas pada bab "Metodologi" akan disajikan kembali pada masing-masing subbab.

5.1.1 Hasil pengujian pengaruh keburaman berat bahan penyusun menu terhadap optimalitas (Pengujian A)

Pada pengujian ini, skenario-skenario yang diujikan ditunjukkan dalam Tabel 5.1.

Tabel 5.1 Skenario-skenario "pengujian A"

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

Dari skenario-skenario pada Tabel 5.1, didapatkan hasil pengujian yang dapat disajikan dalam bentuk matriks hasil pengujian pada Tabel 5.2 dan Tabel 5.3.

Tabel 5.2 Matriks hasil "pengujian A" (1)

Scenario	Menu Set																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A.0	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
A.1	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
A.2	Y	-	-	-	Y	-	-	Y	-	-	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
A.3	Y	-	-	-	Y	-	-	-	-	-	Y	-	-	-	-	-	Y	-	Y	-	-	Y	-	-	Y
A.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	Y
A.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Tabel 5.3 Matriks hasil "pengujian A" (2)

Scenario	Menu Set																							
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
A.0	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	Y	Y	-
A.1	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	Y	Y	-
A.2	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	Y	-
A.3	-	-	-	-	-	Y	-	Y	-	-	-	-	-	-	-	-	-	-	Y	-	-	Y	-	-	-
A.4	-	-	-	-	-	Y	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
A.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A.10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

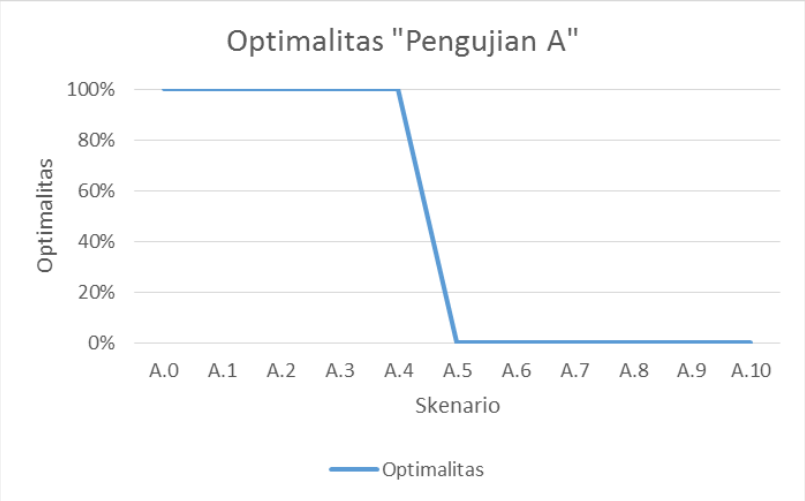
Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Dari matriks hasil pengujian pada Tabel 5.2 dan Tabel 5.3, dapat dibentuk sebuah tabel statistik "Pengujian A" yang disajikan pada Tabel 5.4.

Tabel 5.4 Statistik "pengujian A"

Skenario	Jumlah kasus	Feasible-optimal	Feasible-tak optimal	Tidak ada solusi feasible	Optimalitas
A.0	50	27	0	23	100%
A.1	50	27	0	23	100%
A.2	50	22	0	28	100%
A.3	50	11	0	39	100%
A.4	50	5	0	45	100%
A.5	50	0	0	50	0%
A.6	50	0	0	50	0%
A.7	50	0	0	50	0%
A.8	50	0	0	50	0%
A.9	50	0	0	50	0%
A.10	50	0	0	50	0%

Dari statistik "Pengujian A" pada Tabel 5.4, dapat digambarkan sebuah grafik optimalitas untuk "Pengujian A" seperti pada Gambar 5.1.



Gambar 5.1 Grafik optimalitas "pengujian A"

Grafik optimalitas "Pengujian A" pada Gambar 5.1 menunjukkan bahwa keburaman berat bahan penyusun menu tidak mempengaruhi optimalitas algoritma. Seperti digambarkan dalam grafik tersebut, optimalitas algoritma tetap terjaga pada angka 100% hingga pada skenario di mana untuk semua kasus tidak memiliki solusi *feasible*. Berdasarkan hasil dari "pengujian A" ini, dapat disimpulkan bahwa tidak menjadi masalah seberapa buram data berat bahan penyusun menu yang mungkin dimasukkan pengguna, algoritma tetap dapat menemukan kombinasi paling optimal jika memang masalah tersebut memiliki solusi.

5.1.2 Hasil pengujian pengaruh keburaman harga menu terhadap optimalitas (Pengujian B)

Pada pengujian ini, skenario-skenario yang diujikan ditunjukkan dalam Tabel 5.5.

Tabel 5.5 Skenario-skenario "pengujian B"

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

Dari skenario-skenario pada Tabel 5.5, didapatkan hasil pengujian yang dapat disajikan dalam bentuk matriks hasil pengujian pada Tabel 5.6 dan Tabel 5.7.

Tabel 5.6 Matriks hasil "pengujian B" (1)

Scenario	Menu Set																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
B.0	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
B.1	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
B.2	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
B.3	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
B.4	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
B.5	Y	-	-	-	Y	-	-	Y	-	Y	-	-	-	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	-	Y
B.6	Y	-	-	-	Y	-	-	Y	-	-	-	-	-	Y	Y	-	-	Y	Y	-	Y	Y	-	-	-
B.7	-	-	-	-	Y	-	-	Y	-	-	-	-	-	Y	Y	-	-	Y	Y	-	Y	Y	-	-	-
B.8	-	-	-	-	Y	-	-	Y	-	-	-	-	-	Y	Y	-	-	Y	Y	-	Y	Y	-	-	-
B.9	-	-	-	-	Y	-	-	Y	-	-	-	-	-	Y	Y	-	-	Y	Y	-	Y	Y	-	-	-
B.10	-	-	-	-	Y	-	-	-	-	-	-	-	-	Y	Y	-	-	Y	Y	-	Y	-	-	-	-

Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Tabel 5.7 Matriks hasil "pengujian B" (2)

Scenario	Menu Set																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
B.0	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	Y	Y	-
B.1	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	Y	Y	-
B.2	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
B.3	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	Y	-
B.4	-	Y	-	-	-	Y	-	Y	-	-	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	-	-
B.5	-	Y	-	-	-	Y	-	Y	-	-	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	-	-
B.6	-	Y	-	-	-	Y	-	Y	-	-	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	-	-
B.7	-	Y	-	-	-	Y	-	Y	-	-	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	-	-
B.8	-	Y	-	-	-	Y	-	Y	-	-	Y	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
B.9	-	Y	-	-	-	Y	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
B.10	-	Y	-	-	-	Y	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-

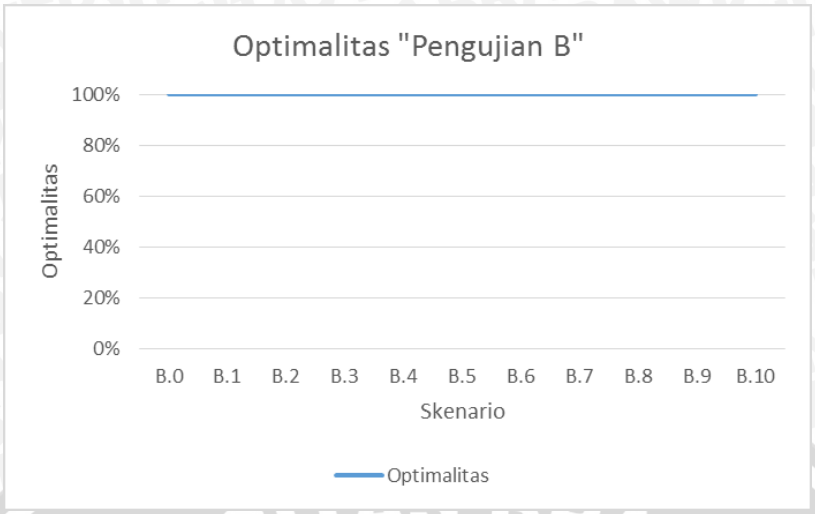
Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Dari matriks hasil pengujian pada Tabel 5.6 dan Tabel 5.7, dapat dibentuk sebuah tabel statistik "Pengujian B" yang disajikan pada Tabel 5.8.

Tabel 5.8 Statistik "pengujian B"

Skenario	Jumlah kasus	Feasible-optimal	Feasible-tak optimal	Tidak ada solusi feasible	Optimalitas
B.0	50	27	0	23	100%
B.1	50	27	0	23	100%
B.2	50	26	0	24	100%
B.3	50	25	0	25	100%
B.4	50	23	0	27	100%
B.5	50	21	0	29	100%
B.6	50	16	0	34	100%
B.7	50	15	0	35	100%
B.8	50	13	0	37	100%
B.9	50	12	0	38	100%
B.10	50	9	0	41	100%

Dari statistik "Pengujian B" pada Tabel 5.8, dapat digambarkan sebuah grafik optimalitas untuk "Pengujian B" seperti pada Gambar 5.2.



Gambar 5.2 Grafik optimalitas "pengujian B"

Grafik optimalitas "Pengujian B" pada Gambar 5.2 menunjukkan bahwa keburaman harga menu tidak mempengaruhi optimalitas algoritma. Seperti digambarkan dalam grafik tersebut, optimalitas algoritma tetap terjaga pada angka 100%. Berdasarkan hasil dari "pengujian B" ini, dapat disimpulkan bahwa tidak menjadi masalah seberapa buram harga yang mungkin dimasukkan pengguna, algoritma tetap dapat menemukan kombinasi paling optimal jika memang masalah tersebut memiliki solusi.

5.1.3 Hasil pengujian pengaruh keburaman batasan gizi pada optimalitas (Pengujian C)

Pada pengujian ini, skenario-skenario yang diujikan ditunjukkan dalam Tabel 5.9.

Tabel 5.9 Skenario-skenario "pengujian C"

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

Dari skenario-skenario pada Tabel 5.9, didapatkan hasil pengujian yang dapat disajikan dalam bentuk matriks hasil pengujian pada Tabel 5.10 dan Tabel 5.11.

Tabel 5.10 Matriks hasil "pengujian C" (1)

Scenario	Menu Set																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C.0	Y	-	-	-	Y	-	-	Y	-	-	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.1	Y	-	-	-	Y	-	-	Y	-	-	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.2	Y	-	-	-	Y	-	-	Y	-	-	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.3	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.4	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.5	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	-	Y	Y	Y	Y	-	Y	Y	-	Y
C.6	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
C.7	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
C.8	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
C.9	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
C.10	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y

Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Tabel 5.11 Matriks hasil "pengujian C" (2)

Scenario	Menu Set																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
C.0	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	Y	-
C.1	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	Y	-
C.2	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	-	-	Y	-	-	Y	-	-	Y	-	Y	-
C.3	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.4	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.5	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.6	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.7	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.8	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.9	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
C.10	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	Y	Y	-

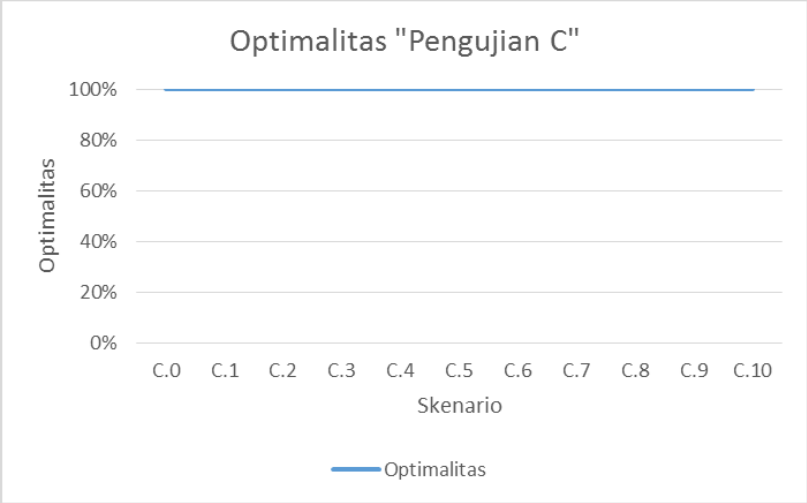
Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Dari matriks hasil pengujian pada Tabel 5.10 dan Tabel 5.11, dapat dibentuk sebuah tabel statistik "Pengujian C" yang disajikan pada Tabel 5.12.

Tabel 5.12 Statistik "pengujian C"

Skenario	Jumlah kasus	Feasible-optimal	Feasible-tak optimal	Tidak ada solusi feasible	Optimalitas
C.0	50	22	0	28	100%
C.1	50	22	0	28	100%
C.2	50	22	0	28	100%
C.3	50	24	0	26	100%
C.4	50	24	0	26	100%
C.5	50	24	0	26	100%
C.6	50	25	0	25	100%
C.7	50	25	0	25	100%
C.8	50	25	0	25	100%
C.9	50	25	0	25	100%
C.10	50	27	0	23	100%

Dari statistik "Pengujian C" pada Tabel 5.12, dapat digambarkan sebuah grafik optimalitas untuk "Pengujian C" seperti pada Gambar 5.3.



Gambar 5.3 Grafik optimalitas "pengujian C"

Grafik optimalitas "Pengujian C" pada Gambar 5.3 menunjukkan bahwa keburaman batasan gizi tidak mempengaruhi optimalitas algoritma. Seperti digambarkan dalam grafik tersebut, optimalitas algoritma tetap terjaga pada angka 100%. Berdasarkan hasil dari "pengujian C" ini, dapat disimpulkan bahwa tidak menjadi masalah seberapa buram batasan gizi yang akan digunakan, algoritma tetap dapat menemukan kombinasi paling optimal jika memang masalah tersebut memiliki solusi. Perlu menjadi catatan bahwa keburaman batasan tidak diperbolehkan melebihi 10%, sesuai dengan panduan yang diberikan oleh pakar gizi.

5.1.4 Hasil pengujian optimalitas untuk kasus nyata (Pengujian D)

Pada pengujian ini, skenario-skenario yang diujikan ditunjukkan dalam Tabel 5.13.

Tabel 5.13 Skenario-skenario "pengujian D"

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

Dari skenario-skenario pada Tabel 5.13, didapatkan hasil pengujian yang dapat disajikan dalam bentuk matriks hasil pengujian pada Tabel 5.14 dan Tabel 5.15.

Tabel 5.14 Matriks hasil "pengujian D" (1)

Scenario	Menu Set																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
D.0	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
D.1	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	-	Y
D.2	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.3	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.4	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.5	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.6	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.7	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.8	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.9	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y
D.10	Y	-	-	-	Y	-	-	Y	-	Y	Y	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y

Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Tabel 5.15 Matriks hasil "pengujian D" (2)

Scenario	Menu Set																								
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
D.0	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.1	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.2	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.3	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.4	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.5	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.6	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.7	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.8	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.9	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-
D.10	-	Y	-	-	-	Y	-	Y	-	Y	Y	-	-	Y	-	Y	-	-	Y	-	-	Y	-	Y	-

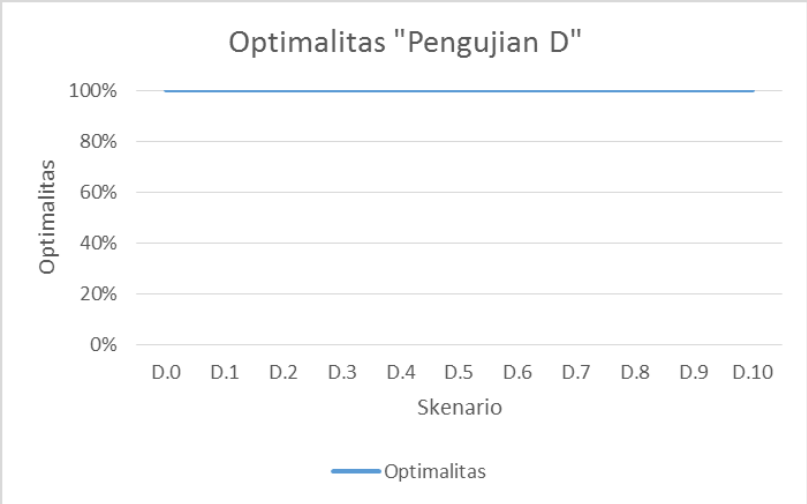
Y = Feasible-optimal
N = Feasible-tak optimal
- = Tidak ada solusi feasible

Dari matriks hasil pengujian pada Tabel 5.14 dan Tabel 5.15, dapat dibentuk sebuah tabel statistik "Pengujian D" yang disajikan pada Tabel 5.16.

Tabel 5.16 Statistik "pengujian D"

Skenario	Jumlah kasus	Feasible-optimal	Feasible-tak optimal	Tidak ada solusi feasible	Optimalitas
D.0	50	25	0	25	100%
D.1	50	25	0	25	100%
D.2	50	26	0	24	100%
D.3	50	26	0	24	100%
D.4	50	26	0	24	100%
D.5	50	26	0	24	100%
D.6	50	26	0	24	100%
D.7	50	26	0	24	100%
D.8	50	26	0	24	100%
D.9	50	26	0	24	100%
D.10	50	26	0	24	100%

Dari statistik "Pengujian D" pada Tabel 5.16, dapat digambarkan sebuah grafik optimalitas untuk "Pengujian D" seperti pada Gambar 5.4.



Gambar 5.4 Grafik optimalitas "pengujian D"

Grafik optimalitas "Pengujian D" pada Gambar 5.4 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.

5.2 Pembahasan

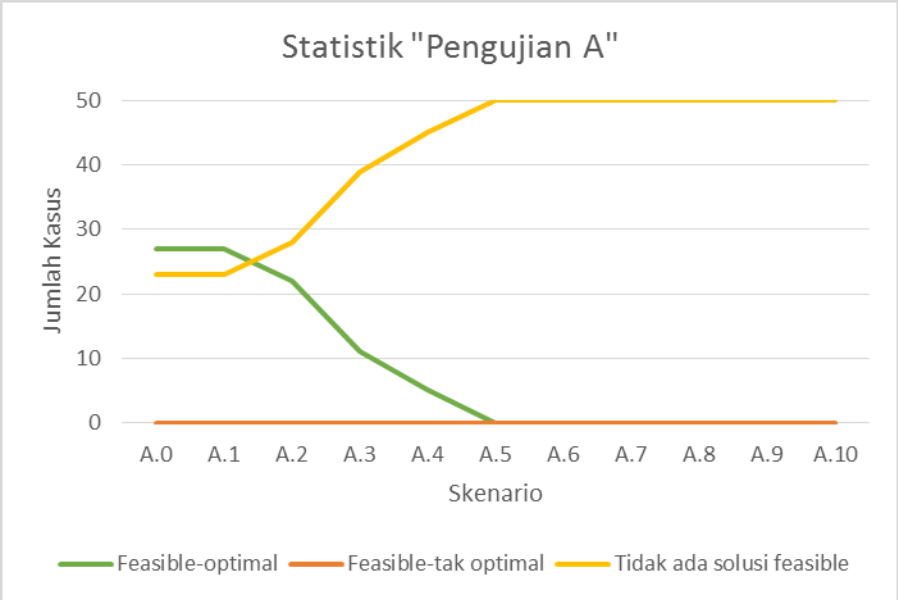
Pada subbab ini akan akan diberikan pembahasan lebih lanjut mengenai apa saja yang dapat ditemukan atau disimpulkan dari data hasil pengujian yang telah disampaikan pada subbab "Hasil Pengujian".

5.2.1 Optimalitas algoritma

Dari grafik optimalitas "Pengujian A" (Gambar 5.1), "Pengujian B" (Gambar 5.2), "Pengujian C" (Gambar 5.3) serta "Pengujian D" (Gambar 5.4), dapat disimpulkan bahwa algoritma FIP dapat menghasilkan solusi yang 100% optimal. Artinya, setiap solusi yang diberikan oleh algoritma FIP adalah solusi dengan harga paling murah dan dengan nilai gizi berada dalam batas-batas yang diizinkan. Pengujian-pengujian tersebut juga menunjukkan bahwa optimalitas algoritma FIP tidak terpengaruh oleh keburaman data, baik data berat bahan penyusun menu, data harga menu, maupun data batasan gizi. Pada setiap skenario yang diujikan, algoritma FIP selalu dapat menemukan solusi optimal jika memang kasus yang diujikan tersebut memiliki solusi.

5.2.2 Pembahasan hasil “pengujian A”

Kembali pada data statistik “Pengujian A” pada Tabel 5.4, data tersebut dapat disajikan dalam bentuk grafik seperti pada Gambar 5.5.



Gambar 5.5 Grafik statistik “pengujian A”

Dari grafik pada Gambar 5.5 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.

Berikut adalah gambaran mengenai bagaimana keburaman data berat bahan penyusun menu berpengaruh pada jumlah solusi *feasible*. Misalnya untuk Menu Set 1, dari 96 kombinasi menu yang mungkin diambil 2 kombinasi sebagai contoh, yakni K1 dan K2 dengan data seperti yang tersaji pada Tabel 5.17.

Tabel 5.17 Contoh kombinasi menu

Kombinasi	Sambal Goreng Ampela	Nasi		Oseng-oseng Kacang Panjang Campur Tempe		Nasi		Brownis Pisang				Jus Wortel Tomat		
	1	1	1/2	1	1/2	1	1/2	4	3	2	1	1	1/2	1/3
K1	✔		✔		✔		✔	✔				✔		
K2	✔		✔		✔		✔	✔					✔	

Setelah dilakukan perhitungan nutrisi pada kombinasi K1 dan K2 untuk skenario A.0 hingga A.5, akan dihasilkan data kandungan Vitamin C seperti yang tersaji pada Tabel 5.18.

Tabel 5.18 Hasil perhitungan vitamin C dari kombinasi K1 dan K2

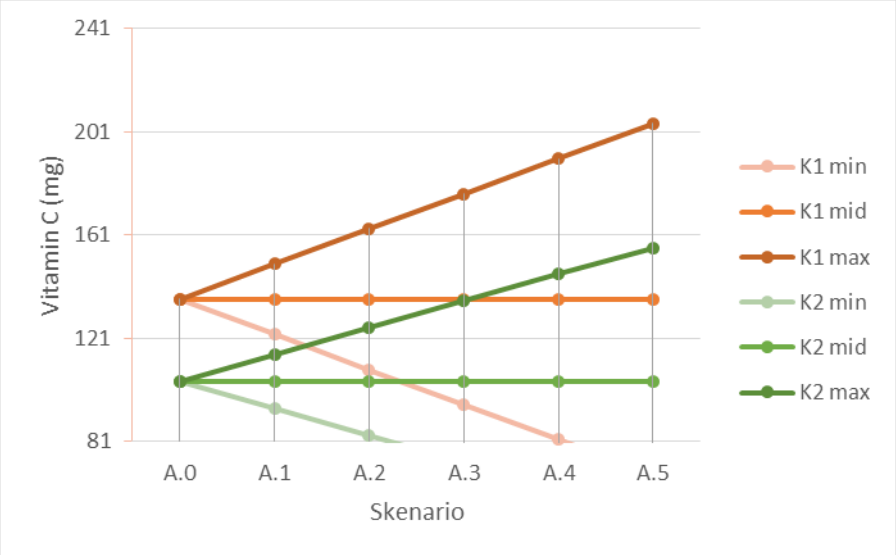
Kombinasi	Skenario	Vitamin C					
		l	s	r	Min.	Mid.	Max.
K1	A.0	0	136.17	0	136.17	136.17	136.17
K1	A.1	13.62	136.17	13.62	122.55	136.17	149.79
K1	A.2	27.23	136.17	27.23	108.94	136.17	163.4
K1	A.3	40.85	136.17	40.85	95.32	136.17	177.02
K1	A.4	54.47	136.17	54.47	81.7	136.17	190.64
K1	A.5	68.08	136.17	68.08	68.09	136.17	204.25
K2	A.0	0	104.13	0	104.13	104.13	104.13
K2	A.1	10.41	104.13	10.41	93.72	104.13	114.54
K2	A.2	20.83	104.13	20.83	83.3	104.13	124.96
K2	A.3	31.24	104.13	31.24	72.89	104.13	135.37
K2	A.4	41.65	104.13	41.65	62.48	104.13	145.78
K2	A.5	52.06	104.13	52.06	52.07	104.13	156.19

Dari Tabel 5.18, dapat disusun sebuah tabel baru yang identik seperti Tabel 5.19.

Tabel 5.19 Kandungan vitamin C kombinasi K1 dan K2

		A.0	A.1	A.2	A.3	A.4	A.5
K1	min	136.17	122.55	108.94	95.32	81.7	68.09
	mid	136.17	136.17	136.17	136.17	136.17	136.17
	max	136.17	149.79	163.4	177.02	190.64	204.25
K2	min	104.13	93.72	83.3	72.89	62.48	52.07
	mid	104.13	104.13	104.13	104.13	104.13	104.13
	max	104.13	114.54	124.96	135.37	145.78	156.19

Tabel 5.19 selanjutnya dapat disajikan dalam bentuk grafik seperti pada Gambar 5.6.



Gambar 5.6 Grafik kandungan vitamin C kombinasi K1 dan K2

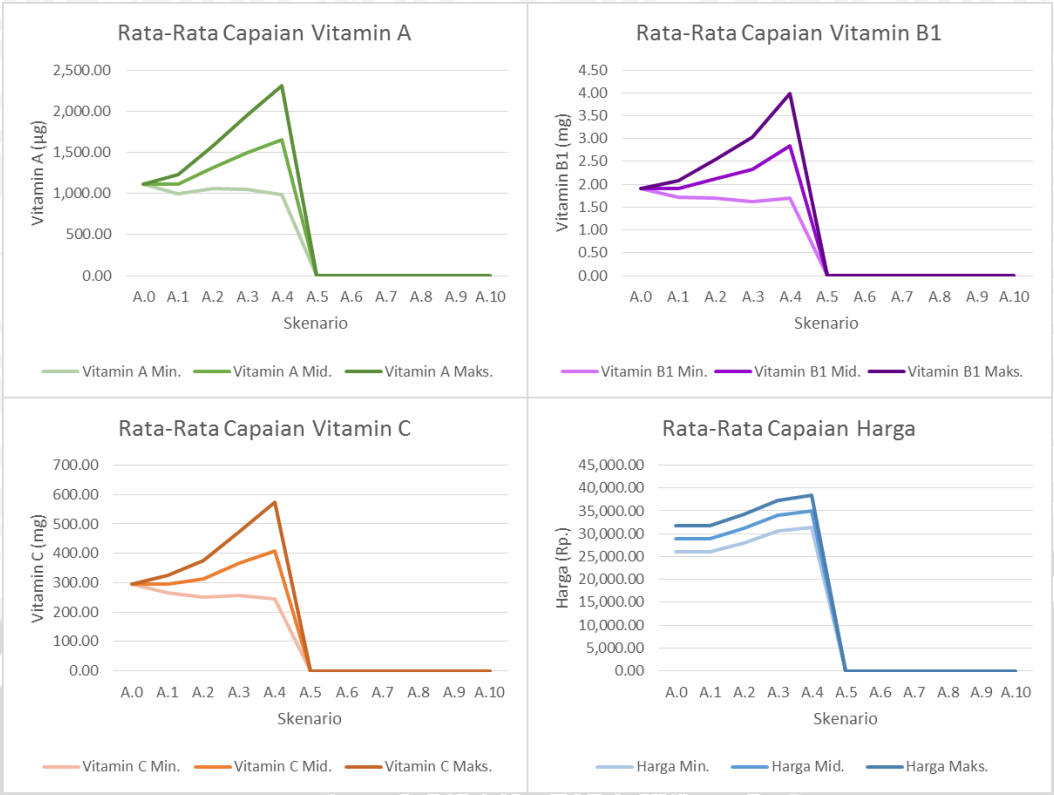
Diketahui batas bawah untuk Vitamin C adalah 90 mg dan batas atasnya adalah 2000 mg. Jika keburaman data pada pengujian ini adalah 10%, maka batas bawah Vitamin C menjadi 81 mg dan batas atasnya 2200 mg. Grafik pada Gambar 5.6 dapat menggambarkan bahwa pada skenario A.0 hingga A.2 kedua kombinasi masih berada dalam batas-batas yang ditentukan. Pada skenario A.3, kombinasi K2 nilai minimumnya, sebesar 72.89 mg, melewati batas bawah yang diizinkan. Vitamin C di sini adalah sebuah *fuzzy number*, yang artinya nilai Vitamin C sesungguhnya dapat berada di mana saja antara nilai minimum dan nilai maksimum. Oleh karena itu, meskipun nilai tengahnya masih dalam batas, kombinasi K2 pada skenario A.3 telah dianggap tidak memenuhi lagi untuk dijadikan sebagai kandidat solusi optimal. Kombinasi K2 telah berubah dari solusi *feasible* di skenario A.0 hingga A.2 menjadi solusi *infeasible* di skenario A.3.

Hal berbeda terjadi pada kombinasi K1. Meskipun nilai minimum dan maksimumnya juga melebar seperti kombinasi K2, namun kombinasi ini masih dapat mempertahankan nilai minimum dan maksimumnya dalam batas yang ditentukan. Kombinasi K1 sampai pada skenario A.4 masih merupakan solusi yang *feasible*. Pada skenario A.5 nilai minimum dari kombinasi ini melewati batas bawah yang diizinkan. Dengan demikian, kombinasi K1 menjadi solusi *infeasible* pada skenario A.5.

Dengan semakin banyaknya solusi yang berubah dari *feasible* menjadi *infeasible*, peluang mendapatkan solusi optimal akan semakin kecil. Hasil pengujian menunjukkan bahwa pada skenario A.5 (keburaman berat bahan penyusun menu = 50%) di semua Menu Set tidak ada satu pun solusi *feasible* sehingga optimalitasnya pun menjadi 0%. Tentu saja hal ini tergantung dari data yang digunakan, namun hasil ini dapat dijadikan sebagai sebuah panduan bahwa keburaman data berat bahan penyusun menu sebaiknya berada di bawah 50%. Angka 10% hingga 20% dirasa cukup untuk mengakomodasi keburaman data berat bahan sekaligus menjaga jumlah solusi *feasible*. Hal lain yang perlu menjadi catatan adalah pada kebanyakan kasus nyata angka keburaman ini tidak akan terlalu ekstrem, di mana keburaman satu bahan yang tinggi mungkin dapat 'diredam' oleh beberapa bahan lain yang keburamannya lebih rendah.

Hasil lain yang didapatkan dari "Pengujian A" adalah bahwa semakin buram data berat bahan penyusun menu, solusi yang dihasilkan cenderung semakin mahal. Hal ini dapat dilihat dari grafik rata-rata capaian dari "Pengujian A" pada Gambar 5.7 dan Gambar 5.8. Angka pada grafik didapatkan dari rata-rata capaian tiap solusi *feasible-optimal* pada sebuah skenario.

Gambar 5.7 Grafik rata-rata capaian "pengujian A" (1)



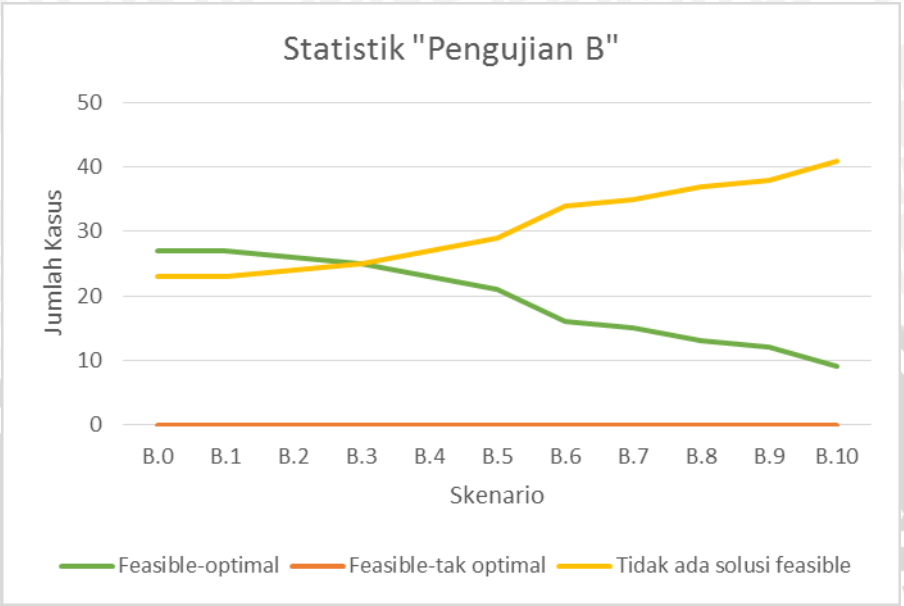
Gambar 5.8 Grafik rata-rata capaian "pengujian A" (2)

Grafik pada Gambar 5.7 dan Gambar 5.8 menunjukkan bahwa semakin buram data berat bahan penyusun menu, kombinasi optimal yang dihasilkan secara garis besar memiliki nilai nutrisi yang semakin baik. Namun, konsekuensi logis dari hal tersebut adalah harga solusi yang semakin mahal. Hal ini juga merupakan efek dari melebarnya nilai minimum dan/atau nilai maksimum yang disebabkan oleh semakin buramnya data berat bahan penyusun menu.

Kembali pada grafik Gambar 5.6, jika dua buah kombinasi mengalami pelebaran yang besarnya sama, maka kombinasi yang paling mungkin untuk tereliminasi (berubah dari solusi *feasible* menjadi *infeasible*) lebih dahulu adalah kombinasi dengan nilai nutrisi lebih rendah. Hal ini karena nilai minimum kombinasi yang memiliki nilai nutrisi lebih rendah akan lebih mudah melewati batas bawah dibandingkan dengan kombinasi dengan nilai nutrisi lebih tinggi. Selain itu, batas atas nutrien biasanya berada jauh di atas nilai normal (misalnya Vitamin C memiliki batas atas 2000 mg), beberapa bahkan tak terbatas, sehingga kecil kemungkinan bahwa kombinasi dengan nilai nutrisi lebih tinggi nilai maksimumnya melebihi batas atas tersebut dan tereliminasi. Inilah yang mengakibatkan pada tingkat keburaman berat bahan penyusun menu yang lebih tinggi, kombinasi-kombinasi yang berpeluang untuk menjadi solusi optimal adalah kombinasi-kombinasi dengan nilai nutrisi yang lebih tinggi, serta dengan harga yang lebih mahal.

5.2.3 Pembahasan hasil "pengujian B"

Kembali pada data statistik "Pengujian B" pada Tabel 5.8, data tersebut dapat disajikan dalam bentuk grafik seperti pada Gambar 5.9.



Gambar 5.9 Grafik statistik "pengujian B"

Dari grafik pada Gambar 5.9 terlihat bahwa semakin buram harga menu mengakibatkan jumlah masalah yang memiliki solusi semakin menurun. Efek ini sama dengan yang terjadi pada "Pengujian A". Perbedaannya adalah bahwa pada pengujian ini yang menjadi penyebab turunnya jumlah solusi *feasible* adalah melebarnya nilai minimum dan/atau maksimum dari harga.

Untuk menjelaskan hal ini, ambil contoh 2 buah kombinasi K1 dan K2 dari 96 kombinasi yang mungkin dari Menu Set 1. Kombinasi-kombinasi tersebut memiliki data seperti yang tersaji pada Tabel 5.20.

Tabel 5.20 Contoh kombinasi menu

Kombinasi	Sambal Goreng Ampela	Nasi		Oseng-oseng Kacang Panjang Campur Tempe		Nasi		Brownis Pisang				Jus Wortel Tomat		
	1	1	1/2	1	1/2	1	1/2	4	3	2	1	1	1/2	1/3
K1	✔		✔		✔		✔		✔				✔	
K2	✔		✔		✔		✔	✔						✔

Perhitungan harga pada kombinasi K1 dan K2 untuk skenario B.0 hingga B.7 akan menghasilkan data harga seperti yang tersaji pada Tabel 5.21.

Tabel 5.21 Hasil perhitungan harga dari kombinasi K1 dan K2

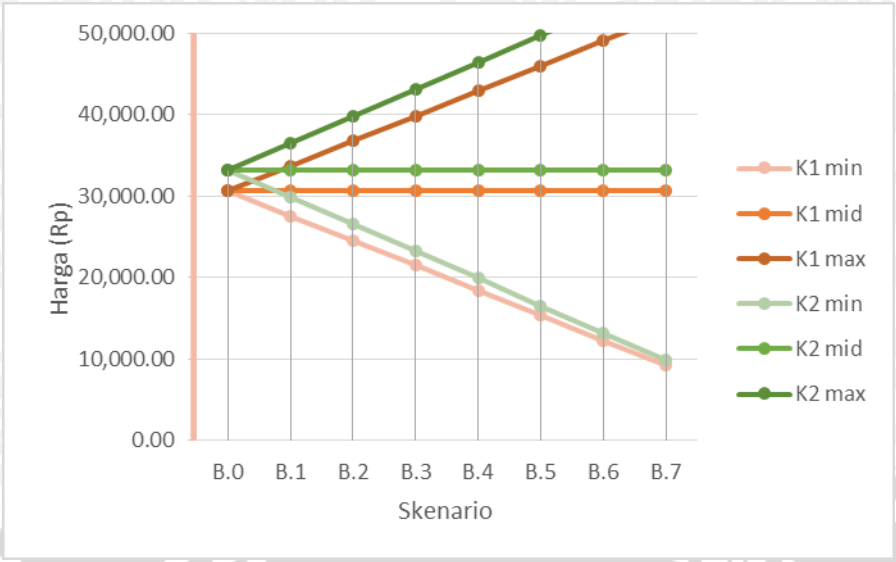
Kombinasi	Skenario	Harga					
		l	s	r	Min.	Mid.	Max.
K1	B.0	0.00	30,650.00	0.00	30,650.00	30,650.00	30,650.00
K1	B.1	3,065.00	30,650.00	3,065.00	27,585.00	30,650.00	33,715.00
K1	B.2	6,130.00	30,650.00	6,130.00	24,520.00	30,650.00	36,780.00
K1	B.3	9,195.00	30,650.00	9,195.00	21,455.00	30,650.00	39,845.00
K1	B.4	12,260.00	30,650.00	12,260.00	18,390.00	30,650.00	42,910.00
K1	B.5	15,325.00	30,650.00	15,325.00	15,325.00	30,650.00	45,975.00
K1	B.6	18,390.00	30,650.00	18,390.00	12,260.00	30,650.00	49,040.00
K1	B.7	21,455.00	30,650.00	21,455.00	9,195.00	30,650.00	52,105.00
K2	B.0	0.00	33,128.30	0.00	33,128.30	33,128.30	33,128.30
K2	B.1	3,312.83	33,128.30	3,312.83	29,815.47	33,128.30	36,441.13
K2	B.2	6,625.66	33,128.30	6,625.66	26,502.64	33,128.30	39,753.96
K2	B.3	9,938.49	33,128.30	9,938.49	23,189.81	33,128.30	43,066.79
K2	B.4	13,251.32	33,128.30	13,251.32	19,876.98	33,128.30	46,379.62
K2	B.5	16,564.15	33,128.30	16,564.15	16,564.15	33,128.30	49,692.45
K2	B.6	19,876.98	33,128.30	19,876.98	13,251.32	33,128.30	53,005.28
K2	B.7	23,189.81	33,128.30	23,189.81	9,938.49	33,128.30	56,318.11

Dari Tabel 5.21, dapat disusun sebuah tabel baru yang identik seperti Tabel 5.22.

Tabel 5.22 Harga kombinasi K1 dan K2

		B.0	B.1	B.2	B.3	B.4	B.5	B.6	B.7
K1	min	30,650.00	27,585.00	24,520.00	21,455.00	18,390.00	15,325.00	12,260.00	9,195.00
	mid	30,650.00	30,650.00	30,650.00	30,650.00	30,650.00	30,650.00	30,650.00	30,650.00
	max	30,650.00	33,715.00	36,780.00	39,845.00	42,910.00	45,975.00	49,040.00	52,105.00
K2	min	33,128.30	29,815.47	26,502.64	23,189.81	19,876.98	16,564.15	13,251.32	9,938.49
	mid	33,128.30	33,128.30	33,128.30	33,128.30	33,128.30	33,128.30	33,128.30	33,128.30
	max	33,128.30	36,441.13	39,753.96	43,066.79	46,379.62	49,692.45	53,005.28	56,318.11

Tabel 5.22 selanjutnya dapat disajikan dalam bentuk grafik seperti pada Gambar 5.10.

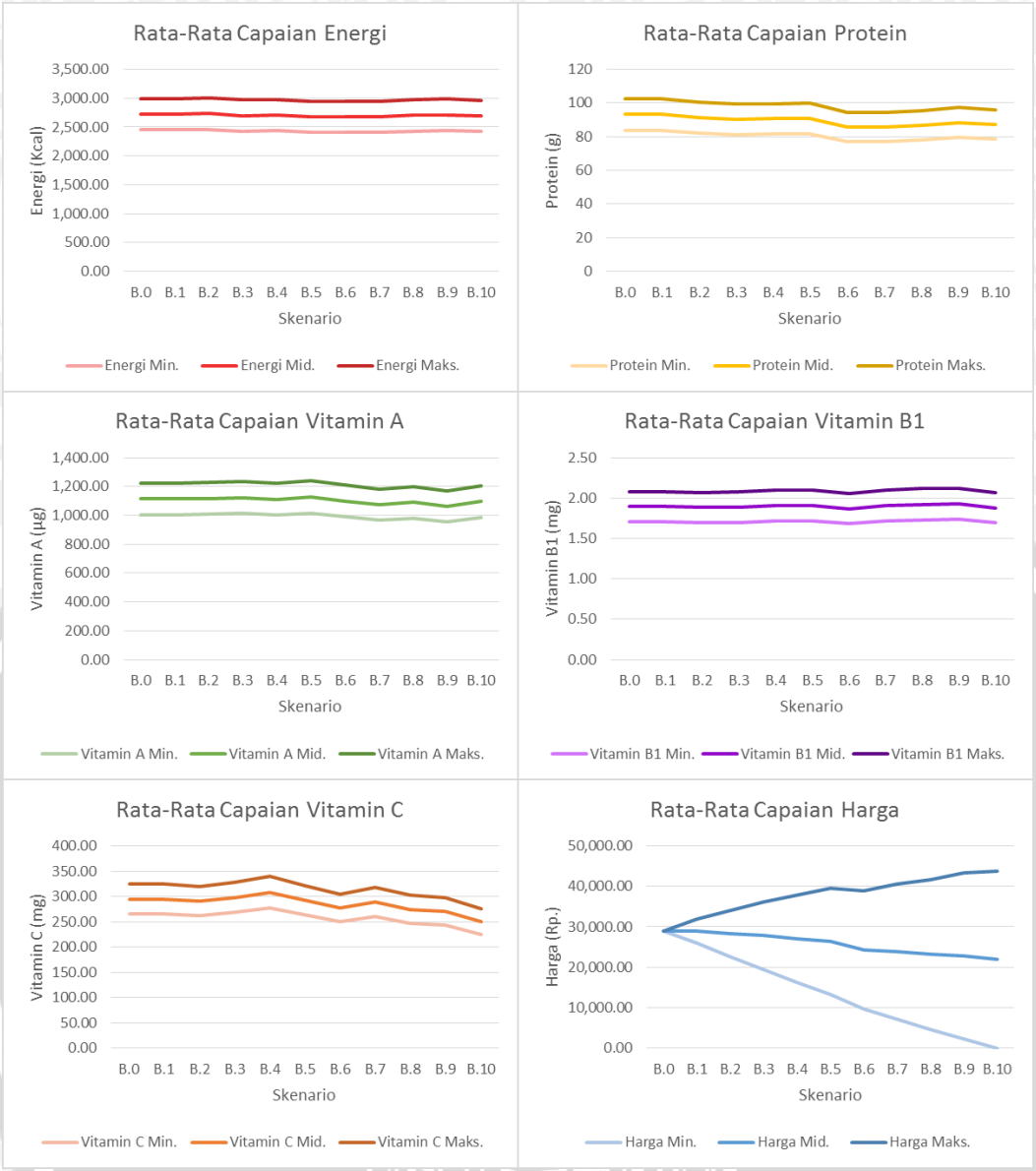


Gambar 5.10 Grafik harga kombinasi K1 dan K2

Batas atas dari harga adalah anggaran yang besarnya Rp. 50.000 dan batas bawahnya adalah Rp. 0. Pada grafik Gambar 5.10 terlihat bahwa pada skenario B.0 hingga B.5 kedua kombinasi masih berada dalam batas-batas yang ditentukan. Pada skenario B.6, kombinasi K2 nilai maksimumnya, sebesar Rp. 53,005.28, melampaui batas atas yang telah ditentukan. Pada skenario ini, K2 telah berubah menjadi solusi *infeasible*. Kombinasi K1 dari skenario B.0 hingga skenario B.6 masih dapat mempertahankan nilai minimum dan nilai maksimumnya dalam batasan-batasan harga. Hingga pada skenario B.7 nilai maksimumnya melebihi batas atas yang ditentukan.

Dari grafik statistik "Pengujian B" pada Gambar 5.9 terlihat bahwa pada pengujian ini masih didapatkan solusi *feasible-optimal* bahkan pada skenario di mana keburaman harga menu 100%. Namun, hal ini tergantung dari data yang digunakan. Sebagai panduan, keburaman harga menu 20% hingga 30% dirasa cukup untuk mengakomodasi keburaman harga menu sekaligus menjaga jumlah solusi *feasible*.

Pada "Pengujian A", rata-rata harga kombinasi menunjukkan *trend* peningkatan seiring bertambahnya keburaman data berat bahan penyusun menu. Pada "Pengujian B" ini *trend* yang ditunjukkan adalah semakin menurun dengan semakin buramnya harga menu. Hal ini dapat dilihat dari grafik rata-rata capaian dari "Pengujian B" pada Gambar 5.11.

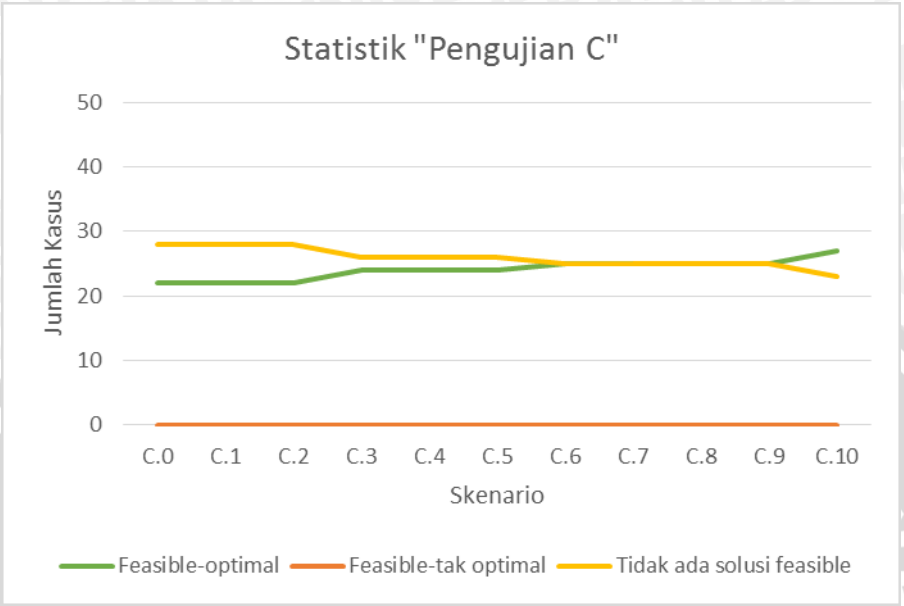


Gambar 5.11 Grafik rata-rata capaian "pengujian B"

Dari grafik pada Gambar 5.11 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.

5.2.4 Pembahasan hasil "Pengujian C"

Kembali pada data statistik "Pengujian C" pada Tabel 5.12, data tersebut dapat disajikan dalam bentuk grafik seperti pada Gambar 5.12.



Gambar 5.12 Grafik statistik "pengujian C"

Dari grafik pada Gambar 5.12 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.

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. Sebagai gambaran, ambil contoh 2 buah kombinasi K1 dan K2 dari 96 kombinasi yang mungkin dari Menu Set 10. Kombinasi-kombinasi tersebut memiliki data seperti yang tersaji pada Tabel 5.23.

Tabel 5.23 Contoh kombinasi menu

Kombinasi	Bubur Ayam		Nasi		Sayur Bening Bayam		Nasi		Sup Ayam		Buah Pepaya		
	1	1/2	1	1/2	1	1/2	1	1/2	1	1/2	3	2	1
K1	✔		✔		✔		✔		✔		✔		
K2	✔			✔		✔		✔	✔			✔	

Setelah dilakukan perhitungan nutrisi pada kombinasi K1 dan K2 untuk skenario C.0 hingga C.6, akan dihasilkan data kandungan energi seperti yang tersaji pada Tabel 5.24.

Tabel 5.24 Hasil perhitungan energi dari kombinasi K1 dan K2

Kombinasi	Skenario	Energi					
		l	s	r	Min.	Mid.	Max.
K1	C.0	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.1	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.2	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.3	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.4	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.5	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K1	C.6	271.22	2712.20	271.22	2440.98	2712.20	2983.42
K2	C.0	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.1	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.2	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.3	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.4	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.5	262.82	2628.24	262.82	2365.42	2628.24	2891.06
K2	C.6	262.82	2628.24	262.82	2365.42	2628.24	2891.06

Dari Tabel 5.24, dapat disusun sebuah tabel baru yang identik seperti Tabel 5.25.

Tabel 5.25 Kandungan energi kombinasi K1 dan K2

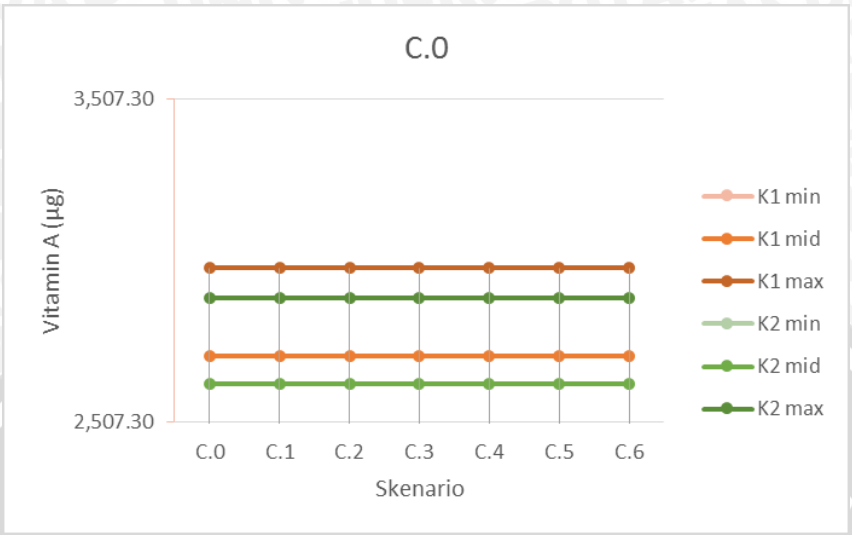
		C.0	C.1	C.2	C.3	C.4	C.5	C.6
K1	min	2440.98	2440.98	2440.98	2440.98	2440.98	2440.98	2440.98
	mid	2712.20	2712.20	2712.20	2712.20	2712.20	2712.20	2712.20
	max	2983.42	2983.42	2983.42	2983.42	2983.42	2983.42	2983.42
K2	min	2365.42	2365.42	2365.42	2365.42	2365.42	2365.42	2365.42
	mid	2628.24	2628.24	2628.24	2628.24	2628.24	2628.24	2628.24
	max	2891.06	2891.06	2891.06	2891.06	2891.06	2891.06	2891.06

Diketahui batas bawah untuk energi adalah sebesar 2507.3 *Kcal* dan energi tidak memiliki batas atas. Dengan demikian, untuk skenario C.0 hingga C.6 diperoleh data batasan seperti yang tersaji pada Tabel 5.26.

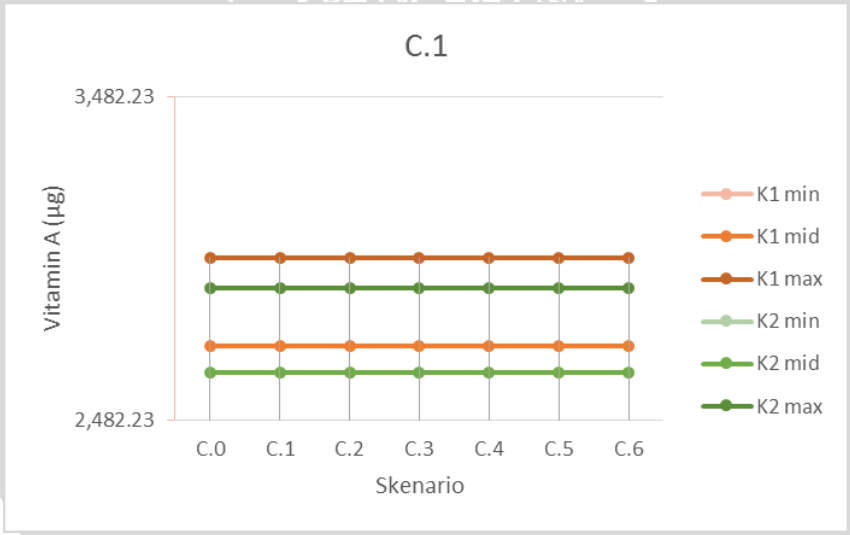
Tabel 5.26 Data batasan energi untuk skenario C.0 hingga C.6

Skenario	Keburaman Batasan (%)	Batas Bawah	Batas Atas
C.0	0	2507.30	Infinity
C.1	1	2482.23	Infinity
C.2	2	2457.15	Infinity
C.3	3	2432.08	Infinity
C.4	4	2407.01	Infinity
C.5	5	2381.94	Infinity
C.6	6	2356.86	Infinity

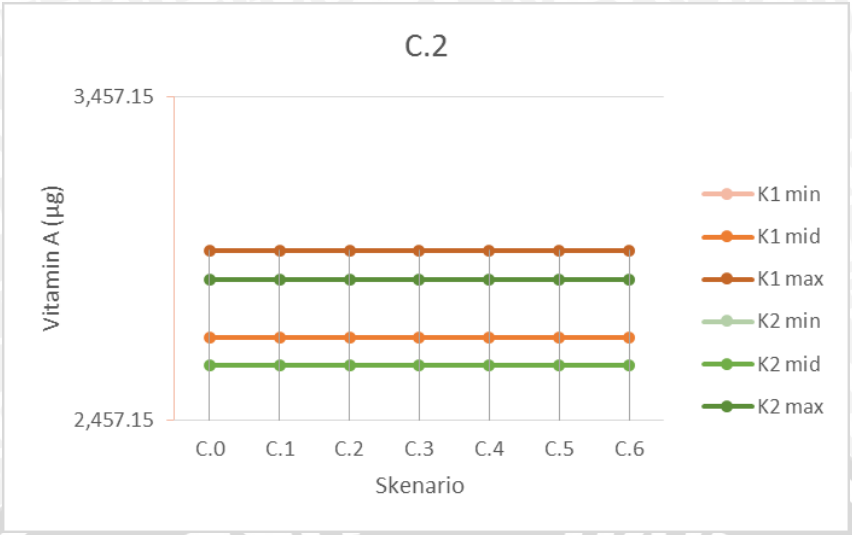
Untuk skenario C.0 hingga C.6, Tabel 5.25 dapat disajikan dalam bentuk grafik secara berurutan pada Gambar 5.13, Gambar 5.14, Gambar 5.15, Gambar 5.16, Gambar 5.17, Gambar 5.18, dan Gambar 5.19.



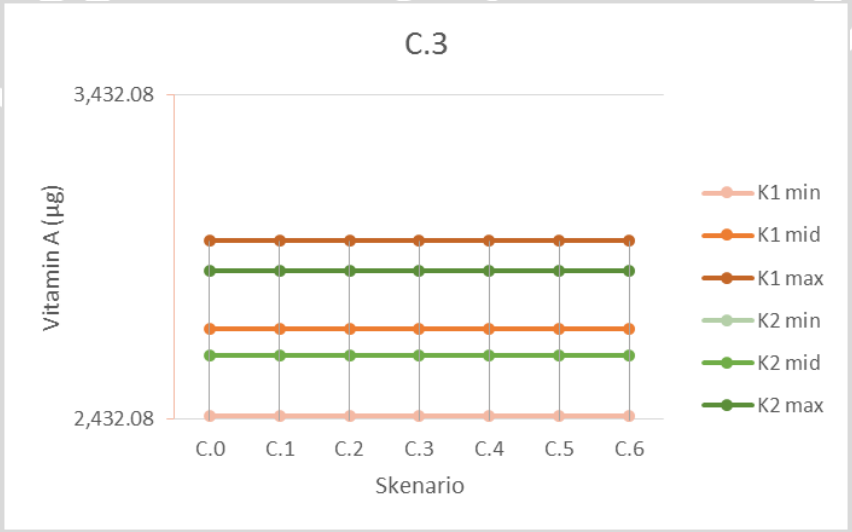
Gambar 5.13 Grafik kandungan energi kombinasi K1 dan K2, skenario C.0



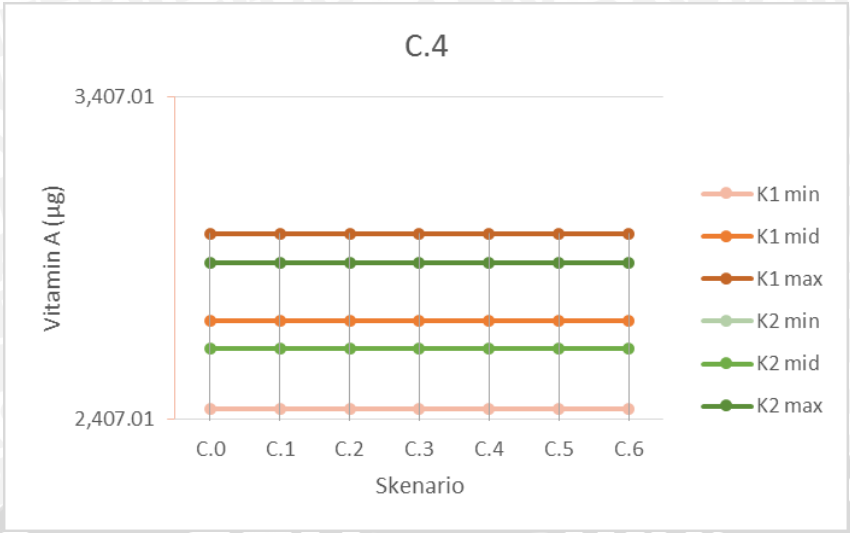
Gambar 5.14 Grafik kandungan energi kombinasi K1 dan K2, skenario C.1



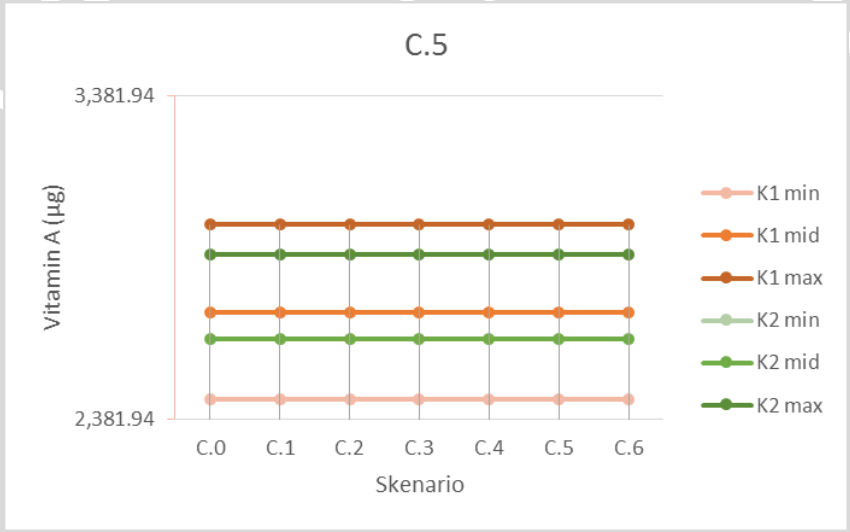
Gambar 5.15 Grafik kandungan energi kombinasi K1 dan K2, skenario C.2



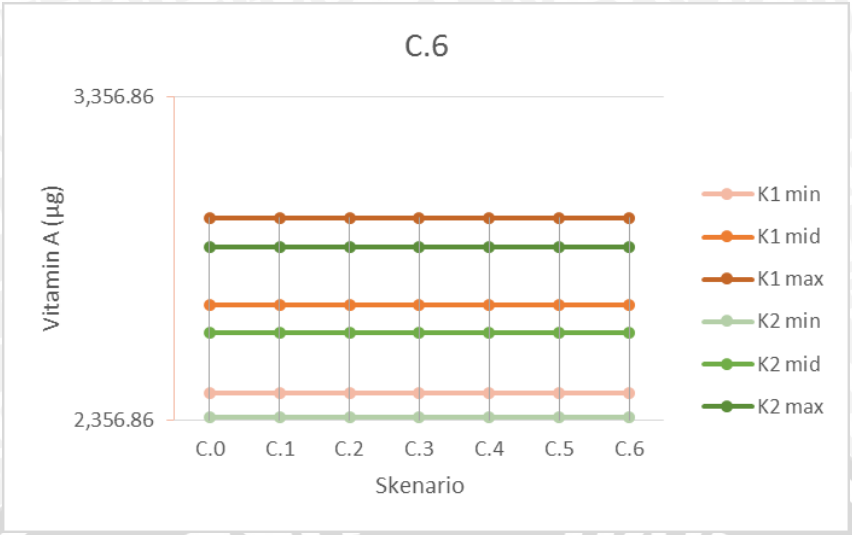
Gambar 5.16 Grafik kandungan energi kombinasi K1 dan K2, skenario C.3



Gambar 5.17 Grafik kandungan energi kombinasi K1 dan K2, skenario C.4



Gambar 5.18 Grafik kandungan energi kombinasi K1 dan K2, skenario C.5

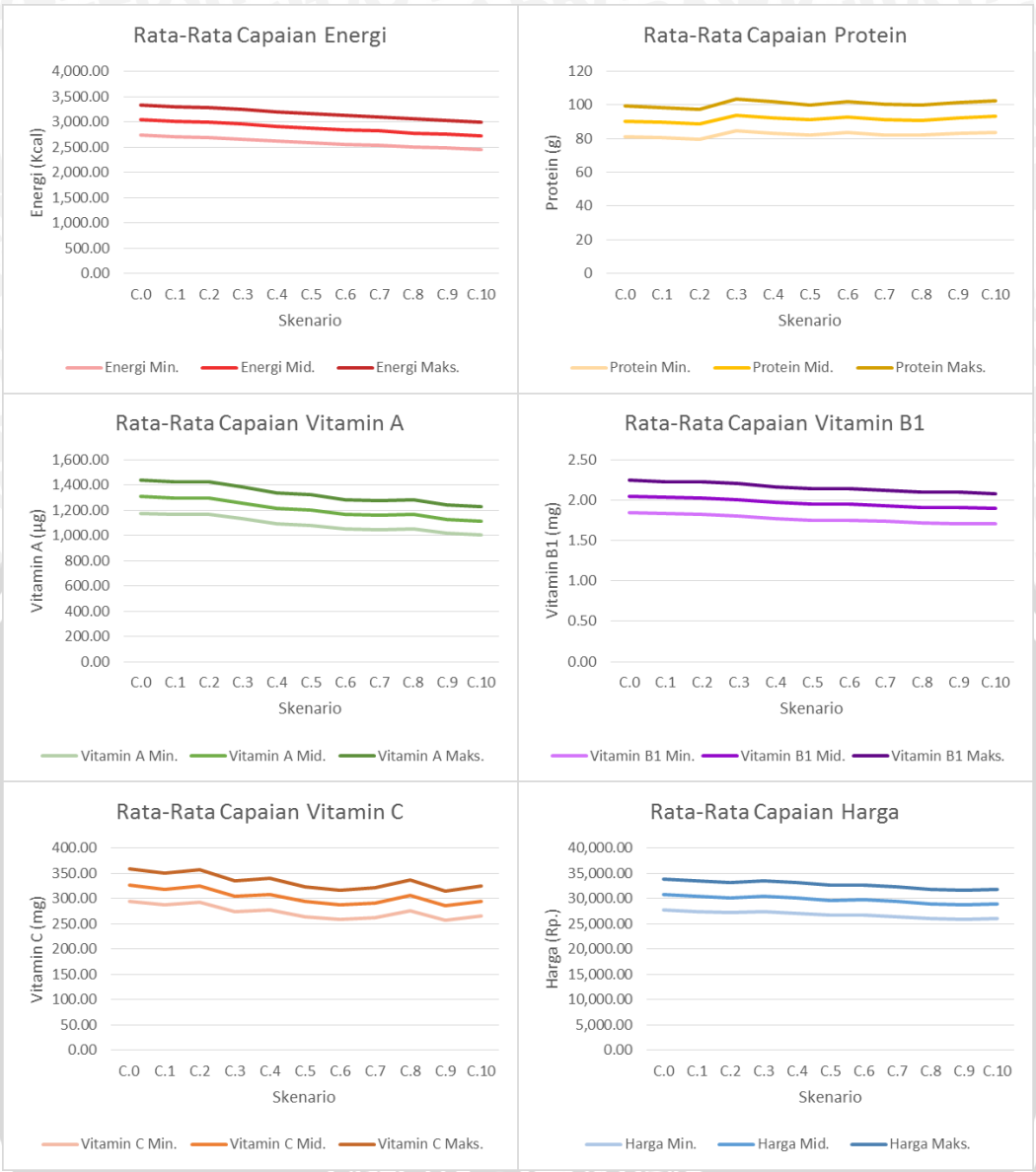


Gambar 5.19 Grafik kandungan energi kombinasi K1 dan K2, skenario C.6

Dari grafik pada Gambar 5.13 hingga Gambar 5.15, terlihat bahwa pada skenario C.0 hingga C.2 baik K1 maupun K2 dianggap sebagai solusi *infeasible* karena nilai minimumnya berada lebih rendah daripada batas bawah yang ditentukan. Pada skenario C.3, dengan semakin buramnya batasan, nilai minimum batasan turun hingga angka 2432.08 *Kcal* memberikan kesempatan bagi kombinasi K1 yang nilai minimum energinya sebesar 2440.98 *Kcal* untuk menjadi kandidat solusi optimal. Pada skenario C.3 (Gambar 5.16) kombinasi K2 masih merupakan solusi *infeasible* karena nilai energi minimumnya hanya sebesar 2365.42 *Kcal*. Hingga pada skenario C.6 (Gambar 5.19) di mana keburaman batasan menyebabkan nilai minimum batasan energi turun ke angka 2356.86 *Kcal*, kombinasi K2 dapat dimasukkan sebagai solusi *feasible* sekaligus kandidat bagi solusi optimal.

Meskipun semakin buram batasan, semakin besar peluang mendapatkan solusi *feasible-optimal*, namun berdasarkan panduan yang diberikan oleh pakar gizi keburaman batasan hanya diperbolehkan hingga sebesar 10%. Dari hasil pengujian ini dapat dikatakan bahwa keburaman batasan 10% merupakan angka yang ideal di mana peluang mendapatkan solusi *feasible-optimal* adalah yang terbesar namun masih dalam batas yang diperbolehkan ditinjau dari aspek ilmu gizi.

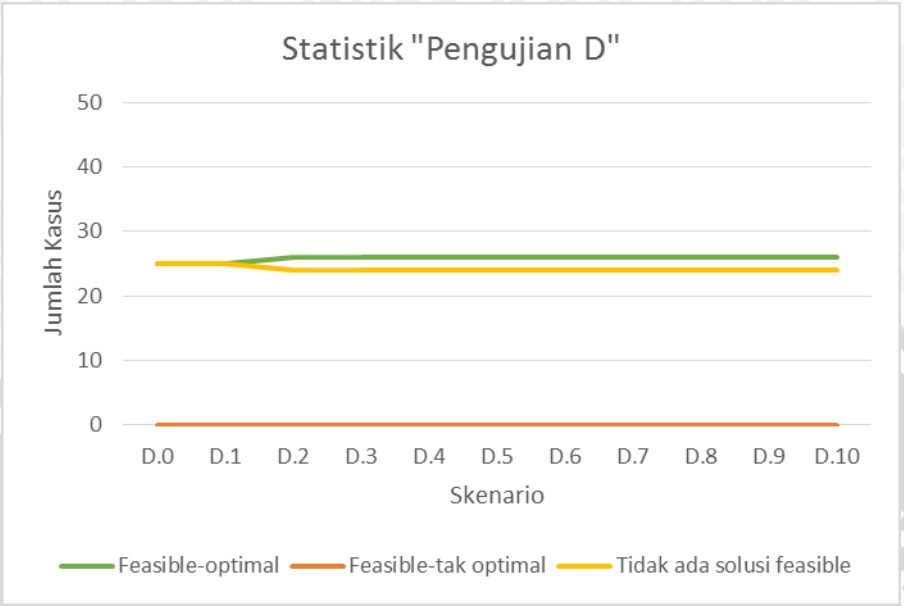
Dari pengujian ini juga didapatkan hasil bahwa semakin buram batasan, kombinasi yang dihasilkan akan semakin murah. Namun demikian, penurunan ini tidak terlalu signifikan mengingat perubahan keburaman batasan juga tidak signifikan. *Trend* penurunan harga ini dapat dilihat pada grafik rata-rata capaian untuk "Pengujian C" pada Gambar 5.20. Penurunan ini dikarenakan oleh penurunan yang terjadi pada batas-batas bawah tiap nutrien, sehingga lebih banyak kombinasi dengan nilai nutrisi lebih rendah (dan dengan harga lebih murah) dapat masuk menjadi kandidat solusi optimal.



Gambar 5.20 Grafik rata-rata capaian "pengujian C"

5.2.5 Pembahasan hasil "pengujian D"

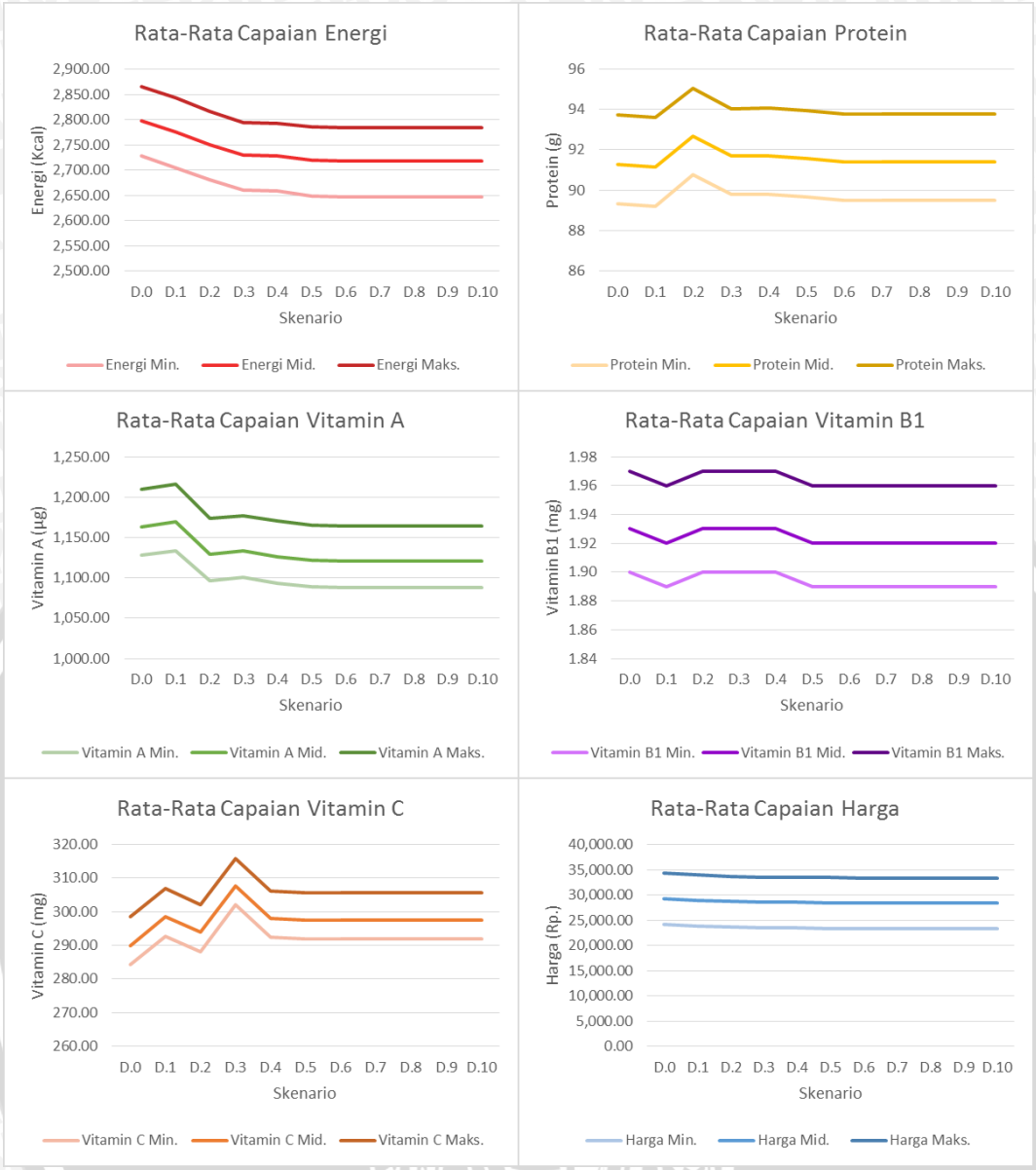
Kembali pada data statistik "Pengujian D" pada Tabel 5.16, data tersebut dapat disajikan dalam bentuk grafik seperti pada Gambar 5.21.



Gambar 5.21 Grafik statistik "pengujian D"

Dari grafik pada Gambar 5.21 terlihat bahwa seiring dengan bertambahnya keburaman batasan terjadi sedikit peningkatan pada jumlah kasus dengan solusi *feasible-optimal*. Hasil ini sama seperti yang didapatkan pada "Pengujian C". Meskipun peningkatan yang terjadi tidak terlalu signifikan, tetapi hasil pengujian ini menunjukkan bahwa pada kasus nyata, di mana keburaman batasan gizi diperbolehkan hingga angka 10%, kemungkinan untuk mendapatkan solusi optimal akan menjadi lebih besar dibandingkan dengan batasan gizi yang nilainya pasti.

Sama seperti hasil dari "Pengujian C", pada pengujian ini terlihat pula bahwa semakin buramnya batasan menyebabkan harga solusi yang diperoleh semakin murah. Namun demikian, penurunan harga tersebut tidak terlalu signifikan mengingat keburaman batasan hanya diperbolehkan hingga angka 10%. *Trend* penurunan harga ini dapat dilihat dari grafik rata-rata capaian harga untuk "Pengujian D" pada Gambar 5.22.

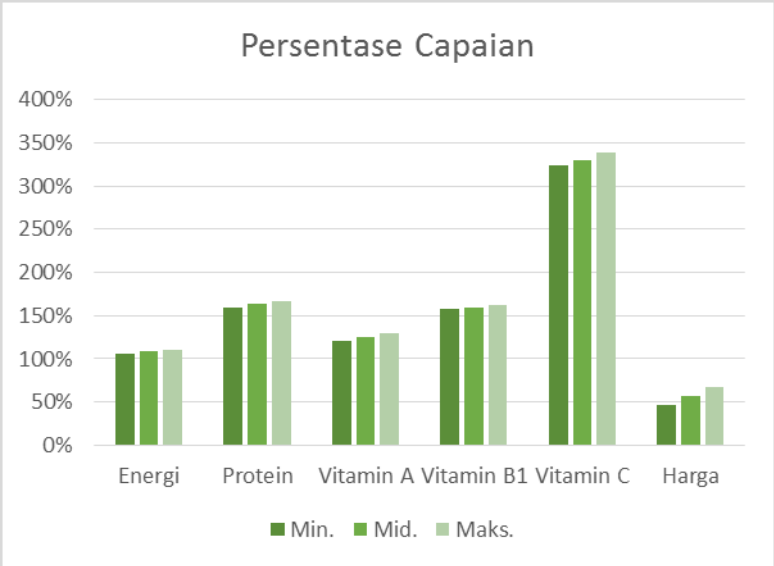


Gambar 5.22 Grafik rata-rata capaian "pengujian D"

Untuk penerapan pada kasus nyata dengan keburaman batasan sebesar 10% (skenario D.10), didapatkan persentase capaian dengan rincian seperti yang tersaji pada Tabel 5.27 dan dapat digambarkan dalam bentuk grafik pada Gambar 5.23.

Tabel 5.27 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 5.23 Grafik persentase capaian penerapan pada kasus nyata

Dari grafik pada Gambar 5.23 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. Untuk Vitamin C, meskipun rata-rata pemenuhannya sekitar 3 kali dari jumlah yang dibutuhkan (rata-rata 305.69 mg), namun masih berada di dalam batasan yang ditentukan. Dengan keburaman batasan 10%, batas bawah untuk vitamin C adalah 810 mg dan batas atasnya adalah 2200 mg. Dari data ini, dapat disimpulkan bahwa solusi-solusi yang diberikan oleh algoritma memiliki nilai gizi yang mencukupi.

Dari sisi harga, solusi-solusi yang dihasilkan juga memenuhi batasan-batasan yang diberikan, yakni tidak melebihi Rp. 50.000. Dari Tabel 5.27 terlihat 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.

BAB 6 PENUTUP

Bab ini berisi ringkasan mengenai hasil yang telah dicapai melalui proses penelitian berupa pernyataan-pernyataan kesimpulan. Bab ini juga akan disertai dengan saran-saran mengenai peluang pengembangan lebih lanjut dari penelitian yang telah dilakukan.

6.1 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. Keburaman data masukan, berupa data berat bahan penyusun menu dan data harga menu, tidak berpengaruh pada optimalitas algoritma. Namun demikian, keburaman tersebut memiliki pengaruh sebagai berikut:
    a. Semakin buram data berat 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%.

6.2 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.
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.

DAFTAR PUSTAKA

Almatsier, S., 2001. *Prinsip Dasar Ilmu Gizi.* Jakarta: Gramedia Pustaka Utama.

Alwanta, M. D. L., 2012. *Penyelesaian Problem Maksimasi pada Metode Simpleks dengan Menggunakan Aplikasi Berbasis Web.* S1. Universitas Negeri Medan. Tersedia di <http://digilib.unimed.ac.id/> [Diakses 2 Mei 2015].

Balintfy, J. L., 1964. Menu Planning by Computer. *Communications of the ACM,* 7(4), pp. 255-259.

Bosch, R. & Trick, M., 2014. Integer Programming. In: *Search Methodologies.* New York: Springer Science+ Business Media, pp. 67-92.

Briend, A. et al., 2003. Linear Programming: A Mathematical Tool for Analyzing and Optimizing Children's Diets During the Complementary Feeding Period. *Journal of Pediatric Gastroenterology and Nutrition,* 36(1), pp. 12-22.

Conforti, P. & D'Amicis, A., 2000. What Is the Cost of a Healthy Diet in Terms of Achieving RDAs?. *Public Health Nutrition,* 3(3), pp. 367-373.

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.

Ford, M., 2006. *Matthew Ford's diet: An Application of the Diet Problem.* S2. Rochester Institute of Technology. Tersedia di <http://scholarworks.rit.edu/theses> [Diakses 21 April 2016].

Fullér, R., 1991. On Product-Sum of Triangular Fuzzy Numbers. *Fuzzy Sets and Systems,* 41(1), pp. 83-87.

Gourvest, H., 2016. *Glpk.js.* [javascript library]. Tersedia di: <http://hgourvest.github.io/glpk.js>.

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).

Lancaster, L. M., 1992. The Evolution of the Diet Model in Managing Food Systems. *Interfaces,* 22(5), pp. 59-68.

Lindo Systems Inc., 2016. *LINGO (15.0).* [program komputer]. Tersedia di: <http://www.lindo.com/> [Diakses 23 April 2016].

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.

Smith, J. C. & Taskin, Z. C., 2008. A Tutorial Guide to Mixed-Integer Programming Models and Solution Techniques. In: G. J. Lim & E. K. Lee, eds. *Optimization in Medicine and Biology.* New York: CRC Press, pp. 521-548.

Sufahani, S. F. & Ismail, Z., 2013. *A Diet Planning Model for Malaysian Boarding School Using Zero-One Integer Programming.*

Suyatno, 2010. *Daftar Komposisi Bahan Makanan (DKBM).* [pdf]. Tersedia di: <http://suyatno.blog.undip.ac.id/files/2010/04/DKBM-Indonesia.pdf> [Diakses 21 April 2016].

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].

Wardlaw, G. M. & Smith, A. M., 2009. *Contemporary Nutrition 7th Edition.* New York: McGraw-Hill Higher Education.

LAMPIRAN A *DIETARY REFERENCE INTAKES* – ELEMEN

Dietary Reference Intakes **(DRIs) : Asupan yang Direkomendasikan** ***(Recommended Intakes)*** **untuk Individu, Elemen**
Food and Nutrition Board, Institute of Medicine, National Academies

Life Stage Group	Calcium (mg/d)	Chromium (µg/d)	Copper (µg/d)	Flouride (mg/d)	Iodine (µg/d)	Iron (mg/d)	Magnesium (mg/d)	Manganese (mg/d)	Molybdenum (µg/d)	Phosphorus (mg/d)	Selenium (µg/d)	Zinc (mg/d)
Infants												
0-6 mo	210*	0.2*	200*	0.01*	110*	0.27*	30*	0.003*	2*	100*	15*	2*
7-12 mo	270*	5.5*	220*	0.5*	130*	**11**	75*	0.6*	3*	275*	20*	**3**
Children												
1-3 y	500*	11*	**340**	0.7*	**90**	**7**	**80**	1.2*	**17**	**460**	**20**	**3**
4-8 y	800*	15*	**440**	1*	**90**	**10**	**130**	1.5*	**22**	**500**	**30**	**5**
Males												
9-13y	1300*	25*	**700**	2*	**120**	**8**	**240**	1.9*	**34**	**1250**	**40**	**8**
14-18y	1300*	35*	**890**	3*	**150**	**11**	**410**	2.2*	**43**	**1250**	**55**	**11**
19-30y	1000*	35*	**900**	4*	**150**	**8**	**400**	2.3*	**45**	**700**	**55**	**11**
31-50y	1000*	35*	**900**	4*	**150**	**8**	**420**	2.3*	**45**	**700**	**55**	**11**
51-70y	1200*	30*	**900**	4*	**150**	**8**	**420**	2.3*	**45**	**700**	**55**	**11**
>70y	1200*	30*	**900**	4*	**150**	**8**	**420**	2.3*	**45**	**700**	**55**	**11**
Females												
9-13y	1300*	21*	**700**	2*	**120**	**8**	**240**	1.6*	**34**	**1250**	**40**	**8**
14-18y	1300*	24*	**890**	3*	**150**	**15**	**360**	1.6*	**43**	**1250**	**55**	**9**
19-30y	1000*	25*	**900**	3*	**150**	**18**	**310**	1.8*	**45**	**700**	**55**	**8**
31-50y	1000*	25*	**900**	3*	**150**	**18**	**320**	1.8*	**45**	**700**	**55**	**8**
51-70y	1200*	20*	**900**	3*	**150**	**8**	**320**	1.8*	**45**	**700**	**55**	**8**
>70y	1200*	20*	**900**	3*	**150**	**8**	**320**	1.8*	**45**	**700**	**55**	**8**
Pregnancy												
≤18y	1300*	29*	**1000**	3*	**220**	**27**	**400**	2.0*	**50**	**1250**	**60**	**12**
19-30y	1000*	30*	**1000**	3*	**220**	**27**	**350**	2.0*	**50**	**700**	**60**	**11**
31-50y	1000*	30*	**1000**	3*	**220**	**27**	**360**	2.0*	**50**	**700**	**60**	**11**
Lactation												
≤18y	1300*	44*	**1300**	3*	**290**	**10**	**360**	2.6*	**50**	**1250**	**70**	**13**
19-30y	1000*	45*	**1300**	3*	**290**	**9**	**310**	2.6*	**50**	**700**	**70**	**12**
31-50y	1000*	45*	**1300**	3*	**290**	**9**	**320**	2.6*	**50**	**700**	**70**	**12**

CATATAN : Tabel ini menyajikan *Recommended Dietary Allowances* (RDA) dalam tulisan **cetak tebal** dan *Adequate Intakes* (AI) dalam tulisan biasa diikuti *asterisk* (*). RDA dan AI keduanya dapat digunakan sebagai acuan untuk asupan individual *(individual intake)* . RDA ditetapkan untuk memenuhi kebutuhan hampir keseluruhan (97 hingga 98 persen) individu dalam kelompok *(group)* . Untuk bayi sehat yang sedang disusui, AI adalah asupan rata-rata. AI untuk kelompok tahap kehidupan *(life stage)* dan gender lain diyakini telah mencakup kebutuhan semua individu dalam kelompok tersebut, namun kurangnya data atau ketidakpastian pada data menyebabkan tidak dapat ditetapkan secara pasti persentase individu yang tercakup oleh asupan tersebut.

LAMPIRAN B *DIETARY REFERENCE INTAKES* – NUTRIEN MAKRO

Dietary Reference Intakes **(DRIs) : Asupan yang Direkomendasikan** ***(Recommended Intakes)*** **untuk Individu, Nutrien Makro**
Food and Nutrition Board, Institute of Medicine, National Academies

Life Stage Group	Carbohydrate (g/d)	Total Fiber (g/d)	Fat (g/d)	Linoleic Acid (g/d)	α-Linoleic Acid (g/d)	Protein[a] (g/d)
Infants						
0-6 mo	60*	ND	31*	4.4*	0.5*	9.1*
7-12 mo	95*	ND	30*	4.6*	0.5*	**13.5**
Children						
1-3 y	**130**	19*	ND[b]	7*	0.7*	**13**
4-8 y	**130**	25*	ND	10*	0.9*	**19**
Males						
9-13y	**130**	31*	ND	12*	1.2*	**34**
14-18y	**130**	38*	ND	16*	1.6*	**52**
19-30y	**130**	38*	ND	17*	1.6*	**56**
31-50y	**130**	38*	ND	17*	1.6*	**56**
51-70y	**130**	30*	ND	14*	1.6*	**56**
>70y	**130**	30*	ND	14*	1.6*	**56**
Females						
9-13y	**130**	26*	ND	10*	1.0*	**34**
14-18y	**130**	26*	ND	11*	1.1*	**46**
19-30y	**130**	25*	ND	12*	1.1*	**46**
31-50y	**130**	25*	ND	12*	1.1*	**46**
51-70y	**130**	21*	ND	11*	1.1*	**46**
>70y	**130**	21*	ND	11*	1.1*	**46**
Pregnancy						
≤18y	**175**	28*	ND	13*	1.4*	**71**
19-30y	**175**	28*	ND	13*	1.4*	**71**
31-50y	**175**	28*	ND	13*	1.4*	**71**
Lactation						
≤18y	**210**	29*	ND	13*	1.3*	**71**
19-30y	**210**	29*	ND	13*	1.3*	**71**
31-50y	**210**	29*	ND	13*	1.3*	**71**

CATATAN : Tabel ini menyajikan *Recommended Dietary Allowances* (RDA) dalam tulisan **cetak tebal** dan *Adequate Intakes* (AI) dalam tulisan biasa diikuti *asterisk* (*). RDA dan AI keduanya dapat digunakan sebagai acuan untuk asupan individual *(individual intake)* . RDA ditetapkan untuk memenuhi kebutuhan hampir keseluruhan (97 hingga 98 persen) individu dalam kelompok *(group)* . Untuk bayi sehat yang sedang disusui, AI adalah asupan rata-rata. AI untuk kelompok tahap kehidupan *(life stage)* dan gender lain diyakini telah mencakup kebutuhan semua individu dalam kelompok tersebut, namun kurangnya data atau ketidakpastian pada data menyebabkan tidak dapat ditetapkan secara pasti persentase individu yang tercakup oleh asupan tersebut.

[a] Berdasarkan 0.8g protein/kg berat badan untuk berat badan rujukan *(reference body weight)*

[b] ND = *Not Determinable* (belum dapat ditentukan) untuk saat ini

LAMPIRAN C *DIETARY REFERENCE INTAKES* – VITAMIN

Dietary Reference Intakes **(DRIs) : Asupan yang Direkomendasikan** ***(Recommended Intakes)*** **untuk Individu, Vitamin**
Food and Nutrition Board, Institute of Medicine, National Academies

Life Stage Group	Vitamin A[a] (µg/d)	Vitamin C (mg/d)	Vitamin D[b,c] (µg/d)	Vitamin E[d] (mg/d)	Vitamin K (µg/d)	Thiamin (mg/d)	Riboflavin (mg/d)	Niacin[e] (mg/d)	Vitamin B-6 (mg/d)	Folate[f] (µg/d)	Vitamin B-12 (µg/d)	Pantothenic Acid (mg/d)	Biotin (µg/d)	Cholin[g] (mg/d)
Infants														
0-6 mo	400*	40*	5*	4*	2.0*	0.2*	0.3*	2*	0.1*	65*	0.4*	1.7*	5*	125*
7-12 mo	500*	50*	5*	5*	2.5*	0.3*	0.4*	4*	0.3*	80*	0.5*	1.8*	6*	150*
Children														
1-3 y	**300**	**15**	5*	**6**	30*	**0.5**	**0.5**	**6**	**0.5**	**150**	**0.9**	2*	8*	200*
4-8 y	**400**	**25**	5*	**7**	55*	**0.6**	**0.6**	**8**	**0.6**	**200**	**1.2**	3*	12*	250*
Males														
9-13y	**600**	**45**	5*	**11**	60*	**0.9**	**0.9**	**12**	**1.0**	**300**	**1.8**	4*	20*	375*
14-18y	**900**	**75**	5*	**15**	75*	**1.2**	**1.3**	**16**	**1.3**	**400**	**2.4**	5*	25*	550*
19-30y	**900**	**90**	5*	**15**	120*	**1.2**	**1.3**	**16**	**1.3**	**400**	**2.4**	5*	30*	550*
31-50y	**900**	**90**	5*	**15**	120*	**1.2**	**1.3**	**16**	**1.3**	**400**	**2.4**	5*	30*	550*
51-70y	**900**	**90**	10*	**15**	120*	**1.2**	**1.3**	**16**	**1.7**	**400**	**2.4**[h]	5*	30*	550*
>70y	**900**	**90**	15*	**15**	120*	**1.2**	**1.3**	**16**	**1.7**	**400**	**2.4**[h]	5*	30*	550*
Females														
9-13y	**600**	**45**	5*	**11**	60*	**0.9**	**0.9**	**12**	**1.0**	**300**	**1.8**	4*	20*	375*
14-18y	**700**	**65**	5*	**15**	75*	**1.0**	**1.0**	**14**	**1.2**	**400**[i]	**2.4**	5*	25*	400*
19-30y	**700**	**75**	5*	**15**	90*	**1.1**	**1.1**	**14**	**1.3**	**400**[i]	**2.4**	5*	30*	425*
31-50y	**700**	**75**	5*	**15**	90*	**1.1**	**1.1**	**14**	**1.3**	**400**[i]	**2.4**	5*	30*	425*
51-70y	**700**	**75**	10*	**15**	90*	**1.1**	**1.1**	**14**	**1.5**	**400**	**2.4**[h]	5*	30*	425*
>70y	**700**	**75**	15*	**15**	90*	**1.1**	**1.1**	**14**	**1.5**	**400**	**2.4**[h]	5*	30*	425*
Pregnancy														
≤18y	**750**	**80**	5*	**15**	75*	**1.4**	**1.4**	**18**	**1.9**	**600**[j]	**2.6**	6*	30*	450*
19-30y	**770**	**85**	5*	**15**	90*	**1.4**	**1.4**	**18**	**1.9**	**600**[j]	**2.6**	6*	30*	450*
31-50y	**770**	**85**	5*	**15**	90*	**1.4**	**1.4**	**18**	**1.9**	**600**[j]	**2.6**	6*	30*	450*
Lactation														
≤18y	**1200**	**115**	5*	**19**	75*	**1.4**	**1.6**	**17**	**2.0**	**500**	**2.8**	7*	35*	550*
19-30y	**1300**	**120**	5*	**19**	90*	**1.4**	**1.6**	**17**	**2.0**	**500**	**2.8**	7*	35*	550*
31-50y	**1300**	**120**	5*	**19**	90*	**1.4**	**1.6**	**17**	**2.0**	**500**	**2.8**	7*	35*	550*

CATATAN : Tabel ini menyajikan *Recommended Dietary Allowances* (RDA) dalam tulisan **cetak tebal** dan *Adequate Intakes* (AI) dalam tulisan biasa diikuti *asterisk* (*). RDA dan AI keduanya dapat digunakan sebagai acuan untuk asupan individual *(individual intake)* . RDA ditetapkan untuk memenuhi kebutuhan hampir keseluruhan (97 hingga 98 persen) individu dalam kelompok *(group)* . Untuk bayi sehat yang sedang disusui, AI adalah asupan rata-rata. AI untuk kelompok tahap kehidupan *(life stage)* dan gender lain diyakini telah mencakup kebutuhan semua individu dalam kelompok tersebut, namun kurangnya data atau ketidakpastian pada data menyebabkan tidak dapat ditetapkan secara pasti persentase individu yang tercakup oleh asupan tersebut.

[a] Sebagai *Retinol Activity Equivalents* (RAE). 1 RAE = 1 µg retinol, 12 µg β-carotene, 24 µg α-carotene, atau 24 µg β-cryptoxanthin. Untuk menghitung RAE dari RE *carotenoid* provitamin A dalam makanan, bagi RE dengan 2. Untuk *preformed* vitamin A dalam makanan atau suplemen dan untuk *carotenoid* provitamin A dalam suplemen 1 RE = 1 RAE.

[b] cholecalciferol. 1 µg cholecalciferol = 40 IU vitamin D.

[c] Dalam ketiadaan paparan sinar matahari yang memadai.

[d] Sebagai α-tocopherol. A-Tocopherol termasuk *RRR*-α-tocopherol, satu-satunya bentuk α-tocopherol yang terdapat secara alami dalam makanan, dan bentuk *2R*-stereoisomeric dari α-tocopherol (*RRR*-, *RSR*-, *RRS*-, dan *RSS*-α-tocopherol) yang terdapat dalam makanan yang diperkaya dan suplemen. Tidak termasuk bentuk *2S*-stereoisomeric dari α-tocopherol (*SRR*-, *SSR*-, *SRS*-, dan *SSS*-α-tocopherol), juga ditemukan dalam makanan yang diperkaya dan suplemen.

[e] Sebagai *Niacin Equivalents* (NE). 1 mg niacin = 60 mg tryptophan; 0-6 bulan (0-6 mo) = *'preformed'* niacin (bukan NE).

[f] Sebagai *Dietary Folate Equivalents* (DFE). 1 DFE = 1µg folate makanan = 0.6 µg folic acid dari makanan yang diperkaya atau sebagai suplemen yang dikonsumsi dengan makanan = 0.5 µg dari suplemen yang diminum saat perut kosong.

[g] Walaupun AI telah ditetapkan untuk choline, ada beberapa data untuk menilai apakah asupan choline dari makanan dibutuhkan pada semua tahap kehidupan, dan memungkinkan bahwa kebutuhan choline dapat dipenuhi dengan *endogenous synthesis* pada sebagian dari tahap-tahap tersebut.

[h] Karena 10 hingga 30 persen dari orang-orang lanjut usia mungkin gagal menyerap B-12 yang terikat dalam makanan *(food-bound B-12)* , disarankan bagi mereka yang berusia di atas 50 tahun memenuhi RDA mereka utamanya dengan mengonsumsi makanan yang diperkaya dengan B-12 atau suplemen yang mengandung B-12.

[i] Dalam sedikit bukti yang mengaitkan asupan folate dengan kelainan tabung saraf *(neural tube defects)* pada janin, disarankan bahwa semua wanita yang memiliki kemampuan untuk hamil mengonsumsi 400 µg dari suplemen atau makanan yang diperkaya sebagai tambahan untuk asupan folate makanan dari diet yang bervariasi.

[j] Diasumsikan bahwa wanita akan terus mengonsumsi 400 µg dari suplemen atau makanan yang diperkaya hingga kehamilan mereka dikonfirmasi dan mereka memasuki perawatan pra-kelahiran, yang mana biasanya terjadi setelah akhir dari masa *periconceptional (periconceptional period)* —waktu kritis untuk pembentukan tabung saraf *(neural tube)*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

LAMPIRAN D *UPPER LEVEL* - VITAMIN

Dietary Reference Intakes **(DRIs) : Batas Atas Asupan yang Masih Dapat Ditoleransi *(Tolerable Upper Intake Levels / UL [a]),* Vitamin**
Food and Nutrition Board, Institute of Medicine, National Academies

Life Stage Group	Vitamin A (µg/d)[b]	Vitamin C (mg/d)	Vitamin D (µg/d)	Vitamin E (mg/d)[c,d]	Vitamin K	Thiamin	Riboflavin	Niacin (mg/d)[d]	Vitamin B-6 (mg/d)	Folate (µg/d)[d]	Vitamin B-12	Pantothenic Acid	Biotin	Choline (g/d)	Carotenoids[e]
Infants															
0-6 mo	600	ND	25	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND
7-12 mo	600	ND	25	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND	ND
Children															
1-3 y	600	400	50	200	ND	ND	ND	10	30	300	ND	ND	ND	1.0	ND
4-8 y	900	650	50	300	ND	ND	ND	15	40	400	ND	ND	ND	1.0	ND
Males, Females															
9-13y	1700	1200	50	600	ND	ND	ND	20	60	600	ND	ND	ND	2.0	ND
14-18y	2800	1800	50	800	ND	ND	ND	30	80	800	ND	ND	ND	3.0	ND
19-70y	3000	2000	50	1000	ND	ND	ND	35	100	1000	ND	ND	ND	3.5	ND
>70y	3000	2000	50	1000	ND	ND	ND	35	100	1000	ND	ND	ND	3.5	ND
Pregnancy															
≤18y	2800	1800	50	800	ND	ND	ND	30	80	800	ND	ND	ND	3.0	ND
19-50y	3000	2000	50	1000	ND	ND	ND	35	100	1000	ND	ND	ND	3.5	ND
Lactation															
≤18y	2800	1800	50	800	ND	ND	ND	30	80	800	ND	ND	ND	3.0	ND
19-50y	3000	2000	50	1000	ND	ND	ND	35	100.0	1000	ND	ND	ND	3.5	ND

[a] UL = Level maksimal asupan nutrien harian yang mungkin tidak memiliki resiko akan dampak merugikan. Kecuali ditentukan lain, UL mewakili asupan total dari makanan, air, dan suplemen. Dikarenakan kurangnya data, UL tidak dapat ditetapkan untuk vitamin K, thiamin, riboflavin, vitamin B-12, pantothenic acid, biotin, atau carotenoids. Dengan tidak adanya UL, kehati-hatian yang lebih dapat dibenarkan dalam mengonsumsi level di atas asupan yang direkomendasikan.

[b] Sebagai *preformed* vitamin A saja.

[c] Sebagai α-tocopherol; berlaku untuk segala bentuk α-tocopherol suplemental.

[d] UL untuk vitamin E, niacin, dan folate berlaku untuk bentuk sintetis yang didapatkan dari suplemen, makanan yang diperkaya, atau kombinasi keduanya.

[e] Suplemen β-carotene disarankan hanya untuk menyajikan sumber provitamin A untuk individu dengan risiko defisiensi vitamin A.

[f] ND = Tidak dapat ditentukan *(Not Determinable)* dikarenakan kurangnya data akan dampak merugikan pada kelompok usia ini dan hal yang berkaitan dengan kurangnya kemampuan untuk menangani jumlah yang berlabih. Sumber asupan harusnya hanya dari makanan untuk mencegah level asupan yang tinggi.

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

LAMPIRAN E *UPPER LEVEL* - ELEMEN DAN ELEKTROLIT

Dietary Reference Intakes **(DRIs) : Batas Atas Asupan yang Masih Dapat Ditoleransi** ***(Tolerable Upper Intake Levels / UL*** [a] ***),*** **Elemen dan Elektrolit**[b,c]
Food and Nutrition Board, Institute of Medicine, National Academies

Life Stage Group	Arsenic[b]	Boron (mg/d)	Calcium (g/d)	Copper (µg/d)	Flouride (mg/d)	Iodine (µg/d)	Iron (mg/d)	Magnesium (mg/d)[d]	Manganese (mg/d)	Molybdenum (µg/d)	Nickel (mg/d)	Phosphorus (g/d)	Selenium (µg/d)	Vanadium (mg/d)[e]	Zinc (mg/d)	Sodium (mg/d)	Chloride (mg/d)
Infants																	
0-6 mo	ND[f]	ND	ND	ND	0.7	ND	40	ND	ND	ND	ND	ND	45	ND	4	ND	ND
7-12 mo	ND	ND	ND	ND	0.9	ND	40	ND	ND	ND	ND	ND	60	ND	5	ND	ND
Children																	
1-3 y	ND	3	2.5	1000	1.3	200	40	65	2	300	0.2	3	90	ND	7	1500	2300
4-8 y	ND	6	2.5	3000	2.2	300	40	110	3	600	0.3	3	150	ND	12	1900	2900
Males, Females																	
9-13y	ND	11	2.5	5000	10	600	40	350	6	1100	0.6	4	280	ND	23	2200	3400
14-18y	ND	17	2.5	8000	10	900	45	350	9	1700	1.0	4	400	ND	34	2300	3600
19-70y	ND	20	2.5	10000	10	1100	45	350	11	2000	1.0	4	400	1.8	40	2300	3600
>70y	ND	20	2.5	10000	10	1100	45	350	11	2000	1.0	3	400	1.8	40	2300	3600
Pregnancy																	
≤18y	ND	17	2.5	8000	10	900	45	350	9	1700	1.0	3.5	400	ND	34	2300	3600
19-50y	ND	20	2.5	10000	10	1100	45	350	11	2000	1.0	3.5	400	ND	40	2300	3600
Lactation																	
≤18y	ND	17	2.5	8000	10	900	45	350	9.0	1700	1.0	4	400	ND	34	2300	3600
19-50y	ND	20	2.5	10000	10	1100	45	350	11.0	2000	1.0	4	400	ND	40	2300	3600

[a] UL = Level maksimal asupan nutrien harian yang mungkin tidak memiliki resiko akan dampak merugikan. Kecuali ditentukan lain, UL mewakili asupan total dari makanan, air, dan suplemen. Dikarenakan kurangnya data, UL tidak dapat ditetapkan untuk arsenic, chromium, dan silicon. Dengan tidak adanya UL, kehati-hatian yang lebih dapat dibenarkan dalam mengonsumsi level di atas asupan yang direkomendasikan.

[b] Walaupun UL tidak ditentukan untuk arsenic, tidak ada pembenaran untuk menambahkan arsenic ke dalam makanan atau suplemen.

[c] Walaupun silicon belum terbukti menyebabkan dampak merugikan pada manusia, tidak ada pembenaran untuk menambahkan silicon ke dalam suplemen.

[d] UL untuk magnesium mewakili asupan dari agen farmakologis saja dan tidak termasuk asupan dari makanan dan air.

[e] Walaupun vanadium dalam makanan belum terbukti menyebabkan dampak merugikan pada manusia, tidak ada pembenaran untuk menambahkan vanadium ke dalam makanan dan suplemen vanadium sebaiknya digunakan dengan hati-hati. UL didasarkan pada dampak merugikan bagi hewan laboratorium dan data ini dapat digunakan untuk menetapkan UL bagi orang dewasa tetapi tidak untuk anak-anak dan remaja.

[f] ND = Tidak dapat ditentukan *(Not Determinable)* dikarenakan kurangnya data akan dampak merugikan pada kelompok usia ini dan hal yang berkaitan dengan kurangnya kemampuan untuk menangani jumlah yang berlabih. Sumber asupan harusnya hanya dari makanan untuk mencegah level asupan yang tinggi.

LAMPIRAN F DATA KANDUNGAN GIZI BAHAN MAKANAN – *PROXIMATES*

Data Bahan - Proximates

Sumber Data
US Department of Agriculture, Agricultural Research Service, Nutrient Data Laboratory
USDA National Nutrient Database for Standard Reference, Release 28. Version Current: September 2015.
Internet: http://www.ars.usda.gov/ba/bhnrc/ndl (http://www.ars.usda.gov/ba/bhnrc/ndl)

Id	Tipe	Nama	carbohydrate	energy	fiber	protein	sugar	water
1001	utama	**Mentega** *Butter, salted*	0.06	717	0	0.85	0	15.87
1009	utama	**Keju Cheddar** *Cheese, cheddar*	3.09	404	0	22.87	0	37.02
1084	utama	**Susu Cair** *Milk, lowfat, fluid, 1% milkfat, protein fortified, with added vitamin A and vitamin D*	5.52	48	0	3.93	0	88.74
1090	utama	**Susu Bubuk** *Milk, dry, whole, with added vitamin D*	38.42	496	0	26.32	0	2.47
1095	utama	**Susu Kental Manis** *Milk, canned, condensed, sweetened*	54.4	321	0	7.91	0	27.16
1116	utama	**Yogurt Tawar** *Yogurt, plain, whole milk, 8 grams protein per 8 ounce*	4.66	61	0	3.47	0	87.9
1123	utama	**Telur** *Egg, whole, raw, fresh*	0.72	143	0	12.56	0	76.15
1125	utama	**Kuning Telur** *Egg, yolk, raw, fresh*	3.59	322	0	15.86	0	52.31
1218	utama	**Yogurt** *Yogurt, fruit variety, nonfat, fortified with vitamin D*	19	95	0	4.4	0	75.4
2004	utama	**Daun Salam** *Spices, bay leaf*	74.97	313	0	7.61	0	5.44
2047	utama	**Garam** *Salt, table*	0	0	0	0	0	0.2
2050	utama	**Vanili** *Vanilla extract*	12.65	288	0	0.06	0	52.58
4058	utama	**Minyak Wijen** *Oil, sesame, salad or cooking*	0	884	0	0	0	0
4696	utama	**Margarin** *Margarine, regular, 80% fat, composite, stick, without salt, with added vitamin D*	0.7	717	0	0.16	0	16.52
5023	utama	**Ampela Ayam** *Chicken, gizzard, all classes, raw*	0	94	0	17.66	0	79.33
5062	utama	**Dada Ayam** *Chicken, broiler or fryers, breast, skinless, boneless, meat only, raw*	0	120	0	22.5	0	73.9
6176	utama	**Saus Tiram** *Sauce, oyster, ready-to-serve*	10.92	51	0	1.35	0	80
7956	utama	**Sosis Sapi** *Beef sausage, fresh, cooked*	0.35	332	0	18.21	0	51.12
8402	utama	**Sereal, QUAKER Quick Oats** *Cereals, QUAKER, Quick Oats, Dry*	68.18	371	0	13.7	0	9.37
9004	utama	**Apel** *Apples, raw, without skin*	12.76	48	0	0.27	0	86.67
9037	utama	**Alpukat** *Avocados, raw, all commercial varieties*	8.53	160	0	2	0	73.23
9040	utama	**Pisang** *Bananas, raw*	22.84	89	0	1.09	0	74.91

9052	utama	**Blueberry Jam** *Blueberries, canned, heavy syrup, solids and liquids*	22.06	88	0	0.65	0	76.78
9139	utama	**Jambu Blji** *Guavas, common, raw*	14.32	68	0	2.55	0	80.8
9152	utama	**Air Lemon** *Lemon juice, raw*	6.9	22	0	0.35	0	92.31
9159	utama	**Jeruk Nipis** *Limes, raw*	10.54	30	0	0.7	0	88.26
9176	utama	**Mangga** *Mangos, raw*	14.98	60	0	0.82	0	83.46
9226	utama	**Pepaya** *Papayas, raw*	10.82	43	0	0.47	0	88.06
9322	utama	**Asam Jawa** *Tamarinds, raw*	62.5	239	0	2.8	0	31.4
9428	utama	**Kluwek** *Rowal, raw*	23.9	111	0	2.3	0	71.4
11003	utama	**Bayam** *Amaranth leaves, raw*	4.02	23	0	2.46	0	91.69
11026	utama	**Rebung** *Bamboo shoots, raw*	5.2	27	0	2.6	0	91
11043	utama	**Tauge** *Mung beans, mature seeds, sprouted, raw*	5.94	30	0	3.04	0	90.4
11052	utama	**Buncis** *Beans, snap, green, raw*	6.97	31	0	1.83	0	90.32
11090	utama	**Brokoli** *Broccoli, raw*	6.64	34	0	2.82	0	89.3
11124	utama	**Wortel** *Carrots, raw*	9.58	41	0	0.93	0	88.29
11135	utama	**Kol** *Cauliflower, raw*	4.97	25	0	1.92	0	92.07
11143	utama	**Seledri** *Celery, raw*	2.97	16	0	0.69	0	95.43
11149	utama	**Labu Siam** *Chayote, fruit, raw*	4.51	19	0	0.82	0	94.24
11167	utama	**Jagung Manis** *Corn, sweet, yellow, raw*	18.7	86	0	3.27	0	76.05
11199	utama	**Kacang Panjang** *Yardlong bean, raw*	8.35	47	0	2.8	0	87.85
11206	utama	**Mentimun** *Cucumber, peeled, raw*	2.16	12	0	0.59	0	96.73
11209	utama	**Terong** *Eggplant, raw*	5.88	25	0	0.98	0	92.3
11215	utama	**Bawang Putih** *Garlic, raw*	33.06	149	0	6.36	0	58.58
11216	utama	**Jahe** *Ginger root, raw*	17.77	80	0	1.82	0	78.89
11238	utama	**Jamur Shiitake** *Mushrooms, shiitake, raw*	6.79	34	0	2.24	0	89.74
11282	utama	**Bawang Merah** *Onions, raw*	9.34	40	0	1.1	0	89.11
11294	utama	**Bawang Bombay** *Onions, sweet, raw*	7.55	32	0	0.8	0	91.24

11333	utama	**Paprika Hijau** *Peppers, sweet, green, raw*	4.64	20	0	0.86	0	93.89
11352	utama	**Kentang** *Potatoes, flesh and skin, raw*	17.49	77	0	2.05	0	79.25
11549	utama	**Saus Tomat** *Tomato products, canned, sauce*	5.31	24	0	1.2	0	91.28
11637	utama	**Lobak Putih** *Radishes, white icicle, raw*	2.63	14	0	1.1	0	95.37
11670	utama	**Cabe Rawit** *Peppers, hot chili, green, raw*	9.46	40	0	2	0	87.74
11695	utama	**Tomat** *Tomatoes, orange, raw*	3.18	16	0	1.16	0	94.78
11819	utama	**Cabe Merah** *Peppers, hot chili, red, raw*	8.81	40	0	1.87	0	88.02
11821	utama	**Paprika Merah** *Peppers, sweet, red, raw*	6.03	31	0	0.99	0	92.21
11994	utama	**Kailan** *Broccoli, chinese, raw*	4.67	30	0	1.2	0	92.55
12117	utama	**Santan** *Nuts, coconut milk, raw (liquid expressed from grated meat and water)*	5.54	230	0	2.29	0	67.62
13000	utama	**Daging Sapi** *Beef, grass-fed, strip steaks, lean only, raw*	0	117	0	23.07	0	73.42
14192	utama	**Coklat Bubuk** *Beverages, Cocoa mix, powder*	83.73	398	0	6.67	0	1.5
15270	utama	**Udang** *Crustaceans, shrimp, raw (not previously frozen)*	0	85	0	20.1	0	78.45
16001	utama	**Kacang Merah** *Beans, adzuki, mature seeds, raw*	62.9	329	0	19.87	0	13.44
16080	utama	**Kacang Hijau** *Mung beans, mature seeds, raw*	62.62	347	0	23.86	0	9.05
16087	utama	**Kacang Tanah** *Peanuts, all types, raw*	16.13	567	0	25.8	0	6.5
16114	utama	**Tempe** *Tempeh*	7.64	192	0	20.29	0	59.65
16124	utama	**Kecap** *Soy sauce made from soy (tamari)*	5.57	60	0	10.51	0	66
16427	utama	**Tahu** *Tofu, raw, regular, prepared with calcium sulfate*	1.87	76	0	8.08	0	84.55
18967	utama	**Roti Tawar** *Bread, white wheat*	43.91	238	0	10.66	0	39.6
19078	utama	**Coklat Batang** *Baking chocolate, unsweetened, squares*	28.42	642	0	14.32	0	1.34
19296	utama	**Madu** *Honey*	82.4	304	0	0.3	0	17.1
19335	utama	**Gula** *Sugars, granulated*	99.98	387	0	0	0	0.02
20027	utama	**Tepung Maizena** *Cornstarch*	91.27	381	0	0.26	0	8.32
20054	utama	**Beras Ketan Putih** *Rice, white, glutinous, unenriched, uncooked*	81.68	370	0	6.81	0	10.46
20084	utama	**Terigu** *Wheat flour, white, cake, enriched*	78.03	362	0	8.2	0	12.51

20133	utama	**Bihun** *Rice noodles, dry*	80.18	364	0	5.95	0	11.91
20409	utama	**Mie Telor** *Noodles, egg, dry, unenriched*	71.27	384	0	14.16	0	9.01
20450	utama	**Beras** *Rice, white, medium-grain, raw, unenriched*	79.34	360	0	6.61	0	12.89
20451	utama	**Nasi** *Rice, white, medium-grain, cooked, unenriched*	28.59	130	0	2.38	0	68.61
20646	utama	**Terigu Protein Tinggi** *Wheat flour, white (industrial), 15% protein, bleached, unenriched*	69.88	362	0	15.33	0	12.85
23307	utama	**Iga Sapi** *Beef, Australian, imported, grass-fed, rib, ribeye steak/roast lip-on, boneless, separable lean only*	0	160	0	21.47	0	70.09

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

Data Bahan - Vitamin

Sumber Data
US Department of Agriculture, Agricultural Research Service, Nutrient Data Laboratory
USDA National Nutrient Database for Standard Reference, Release 28. Version Current: September 2015.
Internet: http://www.ars.usda.gov/ba/bhnrc/ndl (http://www.ars.usda.gov/ba/bhnrc/ndl)

Id	Tipe	Nama	vitamin_a	vitamin_b1	vitamin_b12	vitamin_b2	vitamin_b3	vitamin_b6	vitamin_b9	vitamin_c	vitamin_d	vitamin_e	vitamin_k
1001	utama	**Mentega** *Butter, salted*	684	0.005	0	0.034	0.042	0.003	3	0	0	0	0
1009	utama	**Keju Cheddar** *Cheese, cheddar*	330	0.029	0	0.428	0.059	0.066	27	0	0	0	0
1084	utama	**Susu Cair** *Milk, lowfat, fluid, 1% milkfat, protein fortified, with added vitamin A and vitamin D*	61	0.045	0	0.192	0.1	0.05	6	1.2	0	0	0
1090	utama	**Susu Bubuk** *Milk, dry, whole, with added vitamin D*	258	0.283	0	1.205	0.646	0.302	37	8.6	0	0	0
1095	utama	**Susu Kental Manis** *Milk, canned, condensed, sweetened*	74	0.09	0	0.416	0.21	0.051	11	2.6	0	0	0
1116	utama	**Yogurt Tawar** *Yogurt, plain, whole milk, 8 grams protein per 8 ounce*	27	0.029	0	0.142	0.075	0.032	7	0.5	0	0	0
1123	utama	**Telur** *Egg, whole, raw, fresh*	160	0.04	0	0.457	0.075	0.17	47	0	0	0	0
1125	utama	**Kuning Telur** *Egg, yolk, raw, fresh*	381	0.176	0	0.528	0.024	0.35	146	0	0	0	0
1218	utama	**Yogurt** *Yogurt, fruit variety, nonfat, fortified with vitamin D*	2	0.04	0	0.18	0.1	0.04	9	0.7	0	0	0
2004	utama	**Daun Salam** *Spices, bay leaf*	309	0.009	0	0.421	2.005	1.74	180	46.5	0	0	0

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

2047	utama	**Garam** *Salt, table*	0	0	0	0	0	0	0	0	0	0	0
2050	utama	**Vanili** *Vanilla extract*	0	0.011	0	0.095	0.425	0.026	0	0	0	0	0
4058	utama	**Minyak Wijen** *Oil, sesame, salad or cooking*	0	0	0	0	0	0	0	0	0	0	0
4696	utama	**Margarin** *Margarine, regular, 80% fat, composite, stick, without salt, with added vitamin D*	819	0.01	0	0.037	0.023	0.009	1	0.2	0	0	0
5023	utama	**Ampela Ayam** *Chicken, gizzard, all classes, raw*	19	0.028	0	0.231	3.68	0.112	5	3.7	0	0	0
5062	utama	**Dada Ayam** *Chicken, broiler or fryers, breast, skinless, boneless, meat only, raw*	9	0.094	0	0.177	9.6	0.811	9	0	0	0	0
6176	utama	**Saus Tiram** *Sauce, oyster, ready-to-serve*	0	0.01	0	0.124	1.474	0.016	15	0.1	0	0	0
7956	utama	**Sosis Sapi** *Beef sausage, fresh, cooked*	13	0.048	0	0.15	3.6	0.313	3	0	0	0	0
8402	utama	**Sereal, QUAKER Quick Oats** *Cereals, QUAKER, Quick Oats, Dry*	0	0.54	0	0.12	0.82	0.1	32	0	0	0	0
9004	utama	**Apel** *Apples, raw, without skin*	2	0.019	0	0.028	0.091	0.037	0	4	0	0	0
9037	utama	**Alpukat** *Avocados, raw, all commercial varieties*	7	0.067	0	0.13	1.738	0.257	81	10	0	0	0
9040	utama	**Pisang** *Bananas, raw*	3	0.031	0	0.073	0.665	0.367	20	8.7	0	0	0
9052	utama	**Blueberry Jam**	2	0.034	0	0.053	0.113	0.036	2	1.1	0	0	0

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

		Blueberries, canned, heavy syrup, solids and liquids											
9139	utama	**Jambu Blji** *Guavas, common, raw*	31	0.067	0	0.04	1.084	0.11	49	228.3	0	0	0
9152	utama	**Air Lemon** *Lemon juice, raw*	0	0.024	0	0.015	0.091	0.046	20	38.7	0	0	0
9159	utama	**Jeruk Nipis** *Limes, raw*	2	0.03	0	0.02	0.2	0.043	8	29.1	0	0	0
9176	utama	**Mangga** *Mangos, raw*	54	0.028	0	0.038	0.669	0.119	43	36.4	0	0	0
9226	utama	**Pepaya** *Papayas, raw*	47	0.023	0	0.027	0.357	0.038	37	60.9	0	0	0
9322	utama	**Asam Jawa** *Tamarinds, raw*	2	0.428	0	0.152	1.938	0.066	14	3.5	0	0	0
9428	utama	**Kluwek** *Rowal, raw*	19	0	0	0	0	0	0	25.8	0	0	0
11003	utama	**Bayam** *Amaranth leaves, raw*	146	0.027	0	0.158	0.658	0.192	85	43.3	0	0	0
11026	utama	**Rebung** *Bamboo shoots, raw*	1	0.15	0	0.07	0.6	0.24	7	4	0	0	0
11043	utama	**Tauge** *Mung beans, mature seeds, sprouted, raw*	1	0.084	0	0.124	0.749	0.088	61	13.2	0	0	0
11052	utama	**Buncis** *Beans, snap, green, raw*	35	0.082	0	0.104	0.734	0.141	33	12.2	0	0	0
11090	utama	**Brokoli** *Broccoli, raw*	31	0.071	0	0.117	0.639	0.175	63	89.2	0	0	0
11124	utama	**Wortel** *Carrots, raw*	835	0.066	0	0.058	0.983	0.138	19	5.9	0	0	0
11135	utama	**Kol** *Cauliflower, raw*	0	0.05	0	0.06	0.507	0.184	57	48.2	0	0	0
11143	utama	**Seledri** *Celery, raw*	22	0.021	0	0.057	0.32	0.074	36	3.1	0	0	0

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

11149	utama	**Labu Siam** *Chayote, fruit, raw*	0	0.025	0	0.029	0.47	0.076	93	7.7	0	0	0
11167	utama	**Jagung Manis** *Corn, sweet, yellow, raw*	9	0.155	0	0.055	1.77	0.093	42	6.8	0	0	0
11199	utama	**Kacang Panjang** *Yardlong bean, raw*	43	0.107	0	0.11	0.41	0.024	62	18.8	0	0	0
11206	utama	**Mentimun** *Cucumber, peeled, raw*	4	0.031	0	0.025	0.037	0.051	14	3.2	0	0	0
11209	utama	**Terong** *Eggplant, raw*	1	0.039	0	0.037	0.649	0.084	22	2.2	0	0	0
11215	utama	**Bawang Putih** *Garlic, raw*	0	0.2	0	0.11	0.7	1.235	3	31.2	0	0	0
11216	utama	**Jahe** *Ginger root, raw*	0	0.025	0	0.034	0.75	0.16	11	5	0	0	0
11238	utama	**Jamur Shiitake** *Mushrooms, shiitake, raw*	0	0.015	0	0.217	3.877	0.293	13	0	0	0	0
11282	utama	**Bawang Merah** *Onions, raw*	0	0.046	0	0.027	0.116	0.12	19	7.4	0	0	0
11294	utama	**Bawang Bombay** *Onions, sweet, raw*	0	0.041	0	0.02	0.133	0.13	0	4.8	0	0	0
11333	utama	**Paprika Hijau** *Peppers, sweet, green, raw*	18	0.057	0	0.028	0.48	0.224	10	80.4	0	0	0
11352	utama	**Kentang** *Potatoes, flesh and skin, raw*	0	0.081	0	0.032	1.061	0.298	15	19.7	0	0	0
11549	utama	**Saus Tomat** *Tomato products, canned, sauce*	22	0.024	0	0.065	0.991	0.098	9	7	0	0	0
11637	utama	**Lobak Putih** *Radishes, white*	0	0.03	0	0.02	0.3	0.075	14	29	0	0	0

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

		icicle, raw										
11670	utama	**Cabe Rawit** *Peppers, hot chili, green, raw*	59	0.09	0	0.09	0.95	0.278	23	242.5	0	0	0
11695	utama	**Tomat** *Tomatoes, orange, raw*	75	0.046	0	0.034	0.593	0.06	29	16	0	0	0
11819	utama	**Cabe Merah** *Peppers, hot chili, red, raw*	48	0.072	0	0.086	1.244	0.506	23	143.7	0	0	0
11821	utama	**Paprika Merah** *Peppers, sweet, red, raw*	157	0.054	0	0.085	0.979	0.291	46	127.7	0	0	0
11994	utama	**Kailan** *Broccoli, chinese, raw*	86	0.1	0	0.153	0.459	0.074	104	29.6	0	0	0
12117	utama	**Santan** *Nuts, coconut milk, raw (liquid expressed from grated meat and water)*	0	0.026	0	0	0.76	0.033	16	2.8	0	0	0
13000	utama	**Daging Sapi** *Beef, grass-fed, strip steaks, lean only, raw*	0	0.052	0	0.124	6.703	0.651	13	0	0	0	0
14192	utama	**Coklat Bubuk** *Beverages, Cocoa mix, powder*	1	0.096	0	0.565	0.586	0.114	6	0.2	0	0	0
15270	utama	**Udang** *Crustaceans, shrimp, raw (not previously frozen)*	0	0	0	0	0	0	0	0	0	0	0
16001	utama	**Kacang Merah** *Beans, adzuki, mature seeds, raw*	1	0.455	0	0.22	2.63	0.351	622	0	0	0	0
16080	utama	**Kacang Hijau** *Mung beans, mature seeds, raw*	6	0.621	0	0.233	2.251	0.382	625	4.8	0	0	0
16087	utama	**Kacang Tanah** *Peanuts, all types, raw*	0	0.64	0	0.135	12.066	0.348	240	0	0	0	0

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

16114	utama	**Tempe** *Tempeh*	0	0.078	0	0.358	2.64	0.215	24	0	0	0	0
16124	utama	**Kecap** *Soy sauce made from soy (tamari)*	0	0.059	0	0.152	3.951	0.2	18	0	0	0	0
16427	utama	**Tahu** *Tofu, raw, regular, prepared with calcium sulfate*	0	0.081	0	0.052	0.195	0.047	15	0.1	0	0	0
18967	utama	**Roti Tawar** *Bread, white wheat*	0	0.51	0	0.255	4.455	0.08	162	0	0	0	0
19078	utama	**Coklat Batang** *Baking chocolate, unsweetened, squares*	0	0.147	0	0.1	1.355	0.027	28	0	0	0	0
19296	utama	**Madu** *Honey*	0	0	0	0.038	0.121	0.024	2	0.5	0	0	0
19335	utama	**Gula** *Sugars, granulated*	0	0	0	0.019	0	0	0	0	0	0	0
20027	utama	**Tepung Maizena** *Cornstarch*	0	0	0	0	0	0	0	0	0	0	0
20054	utama	**Beras Ketan Putih** *Rice, white, glutinous, unenriched, uncooked*	0	0.18	0	0.055	2.145	0.107	7	0	0	0	0
20084	utama	**Terigu** *Wheat flour, white, cake, enriched*	0	0.892	0	0.43	6.79	0.033	282	0	0	0	0
20133	utama	**Bihun** *Rice noodles, dry*	0	0.031	0	0.017	0.221	0.015	3	0	0	0	0
20409	utama	**Mie Telor** *Noodles, egg, dry, unenriched*	17	0.17	0	0.09	2.1	0.216	29	0	0	0	0
20450	utama	**Beras** *Rice, white, medium-grain, raw, unenriched*	0	0.07	0	0.048	1.6	0.145	9	0	0	0	0

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

20451	utama	**Nasi** *Rice, white, medium-grain, cooked, unenriched*	0	0.02	0	0.016	0.4	0.05	2	0	0	0	0
20646	utama	**Terigu Protein Tinggi** *Wheat flour, white (industrial), 15% protein, bleached, unenriched*	0	0.194	0	0.072	1.198	0.045	31	0	0	0	0
23307	utama	**Iga Sapi** *Beef, Australian, imported, grass-fed, rib, ribeye steak/roast lip-on, boneless, separable lean only*	0	0	0	0	0	0	0	0	0	0	0

LAMPIRAN H DATA KANDUNGAN GIZI BAHAN MAKANAN – MINERAL

Data Bahan - Mineral

Sumber Data
US Department of Agriculture, Agricultural Research Service, Nutrient Data Laboratory
USDA National Nutrient Database for Standard Reference, Release 28. Version Current: September 2015.
Internet: http://www.ars.usda.gov/ba/bhnrc/ndl (http://www.ars.usda.gov/ba/bhnrc/ndl)

Id	Tipe	Nama	calcium	iron	magnesium	phosphorus	potassium	sodium	zinc
1001	utama	**Mentega** *Butter, salted*	24	0.02	2	24	24	643	0.09
1009	utama	**Keju Cheddar** *Cheese, cheddar*	710	0.14	27	455	76	653	3.64
1084	utama	**Susu Cair** *Milk, lowfat, fluid, 1% milkfat, protein fortified, with added vitamin A and vitamin D*	142	0.06	16	111	180	58	0.45
1090	utama	**Susu Bubuk** *Milk, dry, whole, with added vitamin D*	912	0.47	85	776	1,330	371	3.34
1095	utama	**Susu Kental Manis** *Milk, canned, condensed, sweetened*	284	0.19	26	253	371	127	0.94
1116	utama	**Yogurt Tawar** *Yogurt, plain, whole milk, 8 grams protein per 8 ounce*	121	0.05	12	95	155	46	0.59
1123	utama	**Telur** *Egg, whole, raw, fresh*	56	1.75	12	198	138	142	1.29
1125	utama	**Kuning Telur** *Egg, yolk, raw, fresh*	129	2.73	5	390	109	48	2.3
1218	utama	**Yogurt** *Yogurt, fruit variety, nonfat, fortified with vitamin D*	152	0.07	15	119	194	58	0.74
2004	utama	**Daun Salam** *Spices, bay leaf*	834	43	120	113	529	23	3.7
2047	utama	**Garam** *Salt, table*	24	0.33	1	0	8	38,758	0.1
2050	utama	**Vanili** *Vanilla extract*	11	0.12	12	6	148	9	0.11
4058	utama	**Minyak Wijen** *Oil, sesame, salad or cooking*	0	0	0	0	0	0	0
4696	utama	**Margarin** *Margarine, regular, 80% fat, composite, stick, without salt, with added vitamin D*	3	0.06	3	5	18	2	0
5023	utama	**Ampela Ayam** *Chicken, gizzard, all classes, raw*	11	2.49	15	148	237	69	2.72
5062	utama	**Dada Ayam** *Chicken, broiler or fryers, breast, skinless, boneless, meat only, raw*	5	0.37	28	213	334	45	0.68
6176	utama	**Saus Tiram** *Sauce, oyster, ready-to-serve*	32	0.18	4	22	54	2,733	0.09
7956	utama	**Sosis Sapi** *Beef sausage, fresh, cooked*	11	1.57	14	141	258	813	4.38
8402	utama	**Sereal, QUAKER Quick Oats** *Cereals, QUAKER, Quick Oats, Dry*	47	4.64	270	458	358	3	3.2
9004	utama	**Apel** *Apples, raw, without skin*	5	0.07	4	11	90	0	0.05
9037	utama	**Alpukat**	12	0.55	29	52	485	7	0.64

		Avocados, raw, all commercial varieties							
9040	utama	**Pisang** *Bananas, raw*	5	0.26	27	22	358	1	0.15
9052	utama	**Blueberry Jam** *Blueberries, canned, heavy syrup, solids and liquids*	5	0.33	4	10	40	3	0.07
9139	utama	**Jambu Biji** *Guavas, common, raw*	18	0.26	22	40	417	2	0.23
9152	utama	**Air Lemon** *Lemon juice, raw*	6	0.08	6	8	103	1	0.05
9159	utama	**Jeruk Nipis** *Limes, raw*	33	0.6	6	18	102	2	0.11
9176	utama	**Mangga** *Mangos, raw*	11	0.16	10	14	168	1	0.09
9226	utama	**Pepaya** *Papayas, raw*	20	0.25	21	10	182	8	0.08
9322	utama	**Asam Jawa** *Tamarinds, raw*	74	2.8	92	113	628	28	0.1
9428	utama	**Kluwek** *Rowal, raw*	15	2.2	32	52	131	4	0.43
11003	utama	**Bayam** *Amaranth leaves, raw*	215	2.32	55	50	611	20	0.9
11026	utama	**Rebung** *Bamboo shoots, raw*	13	0.5	3	59	533	4	1.1
11043	utama	**Tauge** *Mung beans, mature seeds, sprouted, raw*	13	0.91	21	54	149	6	0.41
11052	utama	**Buncis** *Beans, snap, green, raw*	37	1.03	25	38	211	6	0.24
11090	utama	**Brokoli** *Broccoli, raw*	47	0.73	21	66	316	33	0.41
11124	utama	**Wortel** *Carrots, raw*	33	0.3	12	35	320	69	0.24
11135	utama	**Kol** *Cauliflower, raw*	22	0.42	15	44	299	30	0.27
11143	utama	**Seledri** *Celery, raw*	40	0.2	11	24	260	80	0.13
11149	utama	**Labu Siam** *Chayote, fruit, raw*	17	0.34	12	18	125	2	0.74
11167	utama	**Jagung Manis** *Corn, sweet, yellow, raw*	2	0.52	37	89	270	15	0.46
11199	utama	**Kacang Panjang** *Yardlong bean, raw*	50	0.47	44	59	240	4	0.37
11206	utama	**Mentimun** *Cucumber, peeled, raw*	14	0.22	12	21	136	2	0.17
11209	utama	**Terong** *Eggplant, raw*	9	0.23	14	24	229	2	0.16
11215	utama	**Bawang Putih** *Garlic, raw*	181	1.7	25	153	401	17	1.16
11216	utama	**Jahe** *Ginger root, raw*	16	0.6	43	34	415	13	0.34
11238	utama	**Jamur Shiitake** *Mushrooms, shiitake, raw*	2	0.41	20	112	304	9	1.03

11282	utama	**Bawang Merah** *Onions, raw*	23	0.21	10	29	146	4	0.17
11294	utama	**Bawang Bombay** *Onions, sweet, raw*	20	0.26	9	27	119	8	0.13
11333	utama	**Paprika Hijau** *Peppers, sweet, green, raw*	10	0.34	10	20	175	3	0.13
11352	utama	**Kentang** *Potatoes, flesh and skin, raw*	12	0.81	23	57	425	6	0.3
11549	utama	**Saus Tomat** *Tomato products, canned, sauce*	14	0.96	15	27	297	474	0.22
11637	utama	**Lobak Putih** *Radishes, white icicle, raw*	27	0.8	9	28	280	16	0.13
11670	utama	**Cabe Rawit** *Peppers, hot chili, green, raw*	18	1.2	25	46	340	7	0.3
11695	utama	**Tomat** *Tomatoes, orange, raw*	5	0.47	8	29	212	42	0.14
11819	utama	**Cabe Merah** *Peppers, hot chili, red, raw*	14	1.03	23	43	322	9	0.26
11821	utama	**Paprika Merah** *Peppers, sweet, red, raw*	7	0.43	12	26	211	4	0.25
11994	utama	**Kailan** *Broccoli, chinese, raw*	105	0.59	19	43	274	7	0.41
12117	utama	**Santan** *Nuts, coconut milk, raw (liquid expressed from grated meat and water)*	16	1.64	37	100	263	15	0.67
13000	utama	**Daging Sapi** *Beef, grass-fed, strip steaks, lean only, raw*	9	1.85	23	212	342	55	3.61
14192	utama	**Coklat Bubuk** *Beverages, Cocoa mix, powder*	133	1.19	83	315	712	504	1.46
15270	utama	**Udang** *Crustaceans, shrimp, raw (not previously frozen)*	64	0.52	35	214	264	119	1.34
16001	utama	**Kacang Merah** *Beans, adzuki, mature seeds, raw*	66	4.98	127	381	1,254	5	5.04
16080	utama	**Kacang Hijau** *Mung beans, mature seeds, raw*	132	6.74	189	367	1,246	15	2.68
16087	utama	**Kacang Tanah** *Peanuts, all types, raw*	92	4.58	168	376	705	18	3.27
16114	utama	**Tempe** *Tempeh*	111	2.7	81	266	412	9	1.14
16124	utama	**Kecap** *Soy sauce made from soy (tamari)*	20	2.38	40	130	212	5,586	0.43
16427	utama	**Tahu** *Tofu, raw, regular, prepared with calcium sulfate*	350	5.36	30	97	121	7	0.8
18967	utama	**Roti Tawar** *Bread, white wheat*	684	4.89	26	103	127	478	0.95
19078	utama	**Coklat Batang** *Baking chocolate, unsweetened, squares*	101	17.4	327	400	830	24	9.63
19296	utama	**Madu** *Honey*	6	0.42	2	4	52	4	0.22
19335	utama	**Gula** *Sugars, granulated*	1	0.05	0	0	2	1	0.01

20027	utama	**Tepung Maizena** *Cornstarch*	2	0.47	3	13	3	9	0.06
20054	utama	**Beras Ketan Putih** *Rice, white, glutinous, unenriched, uncooked*	11	1.6	23	71	77	7	1.2
20084	utama	**Terigu** *Wheat flour, white, cake, enriched*	14	7.32	16	85	105	2	0.62
20133	utama	**Bihun** *Rice noodles, dry*	18	0.7	12	153	30	182	0.74
20409	utama	**Mie Telor** *Noodles, egg, dry, unenriched*	35	1.9	58	241	244	21	1.92
20450	utama	**Beras** *Rice, white, medium-grain, raw, unenriched*	9	0.8	35	108	86	1	1.16
20451	utama	**Nasi** *Rice, white, medium-grain, cooked, unenriched*	3	0.2	13	37	29	0	0.42
20646	utama	**Terigu Protein Tinggi** *Wheat flour, white (industrial), 15% protein, bleached, unenriched*	17	1.26	38	128	120	2	1.16
23307	utama	**Iga Sapi** *Beef, Australian, imported, grass-fed, rib, ribeye steak/roast lip-on, boneless, separable lean only*	5	1.88	0	0	0	65	0

LAMPIRAN I DATA BAHAN TAMBAHAN

Data Bahan Tambahan / Bumbu

Catatan : Bahan-bahan berikut tidak diikutsertakan dalam perhitungan (semua nilai nutrisinya dianggap 0).

Id	Nama	Kategori
1	**Kencur**	bumbu
2	**Lengkuas**	bumbu
3	**Penyedap Rasa**	tambahan
4	**Minyak Goreng**	lain-lain
5	**Air**	lain-lain
6	**Temu Kunci**	bumbu
7	**Kaldu Blok**	tambahan
8	**Es Batu**	lain-lain
9	**Serai**	bumbu
10	**Laos**	bumbu
11	**Daun Jeruk**	bumbu
12	**Ketumbar**	bumbu
13	**Kunyit**	bumbu
14	**Jinten**	bumbu
15	**Kemiri**	bumbu
16	**Daun Bawang**	bumbu
17	**Merica**	bumbu
18	**Kaldu Bubuk**	tambahan
19	**Air Gula**	lain-lain
20	**Air Es**	lain-lain
34	**Lada Bubuk**	bumbu
35	**Biji Pala**	bumbu
36	**Bubuk Cabe Korea**	bumbu
37	**Ovalet**	lain-lain
38	**Daun Melinjo**	tambahan
39	**Gula Merah**	bumbu
40	**Terasi**	bumbu
41	**Cakue**	tambahan
42	**Bawang Goreng**	tambahan

43	**Merica Bubuk**	bumbu
44	**Air Dingin**	lain-lain
45	**Pacar Cina**	tambahan
46	**Risol Bihun**	tambahan
47	**Sari Daun Pandan**	lain-lain
48	**Daun Pandan**	lain-lain
49	**Gula Jawa**	bumbu
2030	**Lada Hitam** *Spices, pepper, black*	bumbu
2032	**Lada** *Spices, pepper, white*	bumbu
18371	**Baking Powder** *Leavening agents, baking powder, low-sodium*	bumbu
18375	**Ragi** *Leavening agents, yeast, bakers, active dry*	bumbu

LAMPIRAN J DATA SATUAN BAHAN MAKANAN

Data Satuan Bahan Makanan

Id	Nama	Satuan
1001	**Mentega** *Butter, salted*	• Sendok Makan (.sdm) *Eqv. 14.2 gram* • Sendok Teh (.sdt) *Eqv. 4.73 gram*
1009	**Keju Cheddar** *Cheese, cheddar*	• Iris (1 ons) (.iris) *Eqv. 28 gram*
1084	**Susu Cair** *Milk, lowfat, fluid, 1% milkfat, protein fortified, with added vitamin A and vitamin D*	• Mililiter (.ml) *Eqv. 1 gram* • Cubic Centimeter (.cc) *Eqv. 1 gram*
1090	**Susu Bubuk** *Milk, dry, whole, with added vitamin D*	• Bungkus (.bks) *Eqv. 27 gram*
1095	**Susu Kental Manis** *Milk, canned, condensed, sweetened*	• Sendok Makan (.sdm) *Eqv. 15 gram* • Sachet (.sachet) *Eqv. 40 gram*
1116	**Yogurt Tawar** *Yogurt, plain, whole milk, 8 grams protein per 8 ounce*	• Sendok Makan (.sdm) *Eqv. 15.3 gram*
1123	**Telur** *Egg, whole, raw, fresh*	• Butir (.btr) *Eqv. 44 gram* • Buah (.buah) *Eqv. 44 gram*
1125	**Kuning Telur** *Egg, yolk, raw, fresh*	• Butir (.btr) *Eqv. 17 gram*
1218	**Yogurt** *Yogurt, fruit variety, nonfat, fortified with vitamin D*	• Sendok Makan (.sdm) *Eqv. 15.3 gram*
2004	**Daun Salam** *Spices, bay leaf*	• Lembar (.lbr) *Eqv. 0.6 gram* • Sendok Teh (.sdt) *Eqv. 0.6 gram* • Sendok Makan (.sdm) *Eqv. 1.8 gram*
2047	**Garam** *Salt, table*	• Sendok Teh (.sdt) *Eqv. 6 gram* • Sendok Makan (.sdm) *Eqv. 18 gram*
2050	**Vanili** *Vanilla extract*	• Sendok Teh (.sdt) *Eqv. 4.2 gram*
4058	**Minyak Wijen** *Oil, sesame, salad or cooking*	• Sendok Makan (.sdm) *Eqv. 13.6 gram*
4696	**Margarin** *Margarine, regular, 80% fat, composite, stick, without salt, with added vitamin D*	• Sendok Teh (.sdt) *Eqv. 4.8 gram* • Sendok Makan (.sdm) *Eqv. 14 gram*
5023	**Ampela Ayam** *Chicken, gizzard, all classes, raw*	• Kilogram (.kg) *Eqv. 1000 gram*
5062	**Dada Ayam** *Chicken, broiler or fryers, breast, skinless, boneless, meat only, raw*	• Buah (.buah) *Eqv. 272 gram*
6176	**Saus Tiram** *Sauce, oyster, ready-to-serve*	• Sendok Makan (.sdm) *Eqv. 18 gram*
7956	**Sosis Sapi** *Beef sausage, fresh, cooked*	• Buah (.buah) *Eqv. 43 gram*
8402	**Sereal, QUAKER Quick Oats** *Cereals, QUAKER, Quick Oats, Dry*	• Cup (.cup) *Eqv. 40 gram*
9004	**Apel** *Apples, raw, without skin*	• Buah (.buah) *Eqv. 161 gram*
9037	**Alpukat**	• Buah (.buah) *Eqv. 201 gram*

	Avocados, raw, all commercial varieties	
9040	**Pisang** *Bananas, raw*	• Buah (.buah) *Eqv. 126 gram*
9052	**Blueberry Jam** *Blueberries, canned, heavy syrup, solids and liquids*	• Sendok Teh (.sdt) *Eqv. 5 gram* • Sendok Makan (.sdm) *Eqv. 10.6 gram*
9139	**Jambu Blji** *Guavas, common, raw*	• Buah (.buah) *Eqv. 55 gram*
9152	**Air Lemon** *Lemon juice, raw*	• Sendok Makan (.sdm) *Eqv. 15 gram*
9159	**Jeruk Nipis** *Limes, raw*	• Buah (.buah) *Eqv. 67 gram*
9176	**Mangga** *Mangos, raw*	• Buah (.buah) *Eqv. 336 gram*
9226	**Pepaya** *Papayas, raw*	• Buah Besar (.buah bsr) *Eqv. 781 gram*
9322	**Asam Jawa** *Tamarinds, raw*	• Buah (3" x 1") (.buah) *Eqv. 2 gram*
9428	**Kluwek** *Rowal, raw*	• Buah (.buah) *Eqv. 60 gram*
11003	**Bayam** *Amaranth leaves, raw*	• Ikat (.ikat) *Eqv. 300 gram*
11043	**Tauge** *Mung beans, mature seeds, sprouted, raw*	• Kilogram (.kg) *Eqv. 1000 gram*
11052	**Buncis** *Beans, snap, green, raw*	• Buah (.buah) *Eqv. 55 gram*
11124	**Wortel** *Carrots, raw*	• Buah Sedang (.buah sdg) *Eqv. 61 gram* • Batang (.btg) *Eqv. 61 gram* • Buah (.buah) *Eqv. 61 gram*
11135	**Kol** *Cauliflower, raw*	• Buah (.buah) *Eqv. 588 gram*
11143	**Seledri** *Celery, raw*	• Helai (.helai) *Eqv. 17 gram* • Tangkai (.tangkai) *Eqv. 17 gram* • Batang (.btg) *Eqv. 17 gram*
11149	**Labu Siam** *Chayote, fruit, raw*	• Buah (.buah) *Eqv. 203 gram*
11167	**Jagung Manis** *Corn, sweet, yellow, raw*	• Buah (.buah) *Eqv. 102 gram*
11199	**Kacang Panjang** *Yardlong bean, raw*	• Ikat (20 btg) (.ikat) *Eqv. 240 gram* • Batang (.btg) *Eqv. 12 gram*
11206	**Mentimun** *Cucumber, peeled, raw*	• Buah Sedang (.buah sdg) *Eqv. 201 gram*
11209	**Terong** *Eggplant, raw*	• Buah (.buah) *Eqv. 458 gram*
11215	**Bawang Putih** *Garlic, raw*	• Buah (.buah) *Eqv. 3 gram* • Sendok Makan (.sdm) *Eqv. 5 gram* • Butir (.btr) *Eqv. 3 gram* • Siung (.siung) *Eqv. 3 gram*

11216	**Jahe** *Ginger root, raw*	• Ruas (.ruas) *Eqv. 11 gram* • Ruas Jari (.ruas jari) *Eqv. 11 gram*
11238	**Jamur Shiitake** *Mushrooms, shiitake, raw*	• Buah (.buah) *Eqv. 19 gram*
11282	**Bawang Merah** *Onions, raw*	• Siung (.siung) *Eqv. 9 gram* • Buah (.buah) *Eqv. 9 gram* • Butir (.btr) *Eqv. 9 gram*
11294	**Bawang Bombay** *Onions, sweet, raw*	• Buah (.buah) *Eqv. 331 gram*
11333	**Paprika Hijau** *Peppers, sweet, green, raw*	• Buah (.buah) *Eqv. 119 gram*
11352	**Kentang** *Potatoes, flesh and skin, raw*	• Buah (.buah) *Eqv. 213 gram* • Buah Sedang (.buah sdg) *Eqv. 213 gram* • Buah Besar (.buah bsr) *Eqv. 369 gram*
11549	**Saus Tomat** *Tomato products, canned, sauce*	• Sendok Makan (.sdm) *Eqv. 16.38 gram*
11637	**Lobak Putih** *Radishes, white icicle, raw*	• Buah (.buah) *Eqv. 17 gram*
11670	**Cabe Rawit** *Peppers, hot chili, green, raw*	• Biji (.biji) *Eqv. 45 gram* • Kilogram (.kg) *Eqv. 1000 gram* • Buah (.buah) *Eqv. 45 gram*
11695	**Tomat** *Tomatoes, orange, raw*	• Buah (.buah) *Eqv. 111 gram*
11819	**Cabe Merah** *Peppers, hot chili, red, raw*	• Biji (.biji) *Eqv. 45 gram* • Buah (.buah) *Eqv. 45 gram*
11821	**Paprika Merah** *Peppers, sweet, red, raw*	• Buah (.buah) *Eqv. 119 gram*
12117	**Santan** *Nuts, coconut milk, raw (liquid expressed from grated meat and water)*	• Liter (.ltr) *Eqv. 1000 gram* • Mililiter (.ml) *Eqv. 1 gram*
14192	**Coklat Bubuk** *Beverages, Cocoa mix, powder*	• Sendok Makan (.sdm) *Eqv. 45 gram* • Sendok Teh (.sdt) *Eqv. 28 gram*
16001	**Kacang Merah** *Beans, adzuki, mature seeds, raw*	• Bungkus (500 gr) (.bks) *Eqv. 500 gram* • Genggam (.genggam) *Eqv. 197 gram*
16087	**Kacang Tanah** *Peanuts, all types, raw*	• Kilogram (.kg) *Eqv. 1000 gram*
16114	**Tempe** *Tempeh*	• Buah (.buah) *Eqv. 200 gram*
16124	**Kecap** *Soy sauce made from soy (tamari)*	• Sendok Makan (.sdm) *Eqv. 18 gram* • Sendok Teh (.sdt) *Eqv. 6 gram*
16427	**Tahu** *Tofu, raw, regular, prepared with calcium sulfate*	• Buah (.buah) *Eqv. 116 gram*
18967	**Roti Tawar** *Bread, white wheat*	• Lembar (.lbr) *Eqv. 28 gram*

19296	**Madu** *Honey*	• Sendok Makan (.sdm) *Eqv. 21 gram*
19335	**Gula** *Sugars, granulated*	• Sendok Makan (.sdm) *Eqv. 12.5 gram* • Sendok Teh (.sdt) *Eqv. 4.2 gram*
20027	**Tepung Maizena** *Cornstarch*	• Sendok Teh (.sdt) *Eqv. 3.13 gram*
20409	**Mie Telor** *Noodles, egg, dry, unenriched*	• Bungkus (.bks) *Eqv. 200 gram*
20451	**Nasi** *Rice, white, medium-grain, cooked, unenriched*	• Cup (.cup) *Eqv. 186 gram* • Piring (.prg) *Eqv. 200 gram*
23307	**Iga Sapi** *Beef, Australian, imported, grass-fed, rib, ribeye steak/roast lip-on, boneless, separable lean only*	• Kilogram (.kg) *Eqv. 1000 gram*

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

Data Menu

Catatan: Bilangan buram *(Fuzzy Number)* dinotasikan dengan *Fuzzy(l, s, r)*. Nilai yang direpresentasikan dengan bilangan ini berada dalam range *(s-l)* hingga *(s+r)* dengan nilai yang paling mungkin sebesar *s*. Sebagai contoh, harga Rp. (1000, 5000, 500) memiliki arti bahwa harga berada dalam *range* Rp. 4000 hingga Rp. 5500 dengan harga yang paling mungkin adalah sebesar Rp. 5000.

Id	Nama	Porsi	Porsi Penyajian Standar	Harga Fuzzy	Bahan Utama	Bahan Tambahan / Bumbu	Valid
1	Sambal Goreng Ampela	5	1	(1000, 5700, 500)	• Ampela Ayam : 1 .kg *Fuzzy eqv. (0, 1000, 0) gram* • Wortel : 2 .buah *Fuzzy eqv. (5, 122, 5) gram* • Kentang : 2 .buah *Fuzzy eqv. (10, 426, 10) gram*	• Kencur : secukupnya • Lengkuas : secukupnya • Penyedap Rasa : secukupnya • Minyak Goreng : secukupnya • Kemiri : 2 .btr • Daun Salam : 1 .lbr • Garam : secukupnya • Bawang Putih : 2 .siung • Jahe : secukupnya • Bawang Merah : 3 .siung • Cabe Merah : 4 .buah • Cabe Merah : 2 .buah • Santan : 0 .gr	
2	Nasi	1	1,1/2	(500, 1700, 500)	• Nasi : 1 .prg *Fuzzy eqv. (20, 200, 10) gram*		
3	Oseng-oseng Kacang Panjang Campur Tempe	4	1,1/2	(500, 1400, 1000)	• Saus Tiram : 2 .sdm *Fuzzy eqv. (2, 36, 2) gram* • Kacang Panjang : 100 .gr *Fuzzy eqv. (0, 100, 0) gram* • Tempe : 1 .buah *Fuzzy eqv. (10, 200, 50) gram* • Kecap : 2 .sdm *Fuzzy eqv. (2, 36, 2) gram*	• Minyak Goreng : secukupnya • Air : secukupnya • Merica : secukupnya • Daun Salam : 2 .lbr • Garam : secukupnya • Bawang Putih : 2 .siung • Jahe : 2 .ruas jari	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

						• Bawang Merah : 2 .siung • Bawang Bombay : 1 .buah • Cabe Rawit : 7 .buah • Cabe Merah : 1 .buah	
4	Cah Kailan	4	2,1	(500, 1900, 500)	• Saus Tiram : 3 .sdm *Fuzzy eqv. (5, 54, 5) gram* • Bawang Putih : 3 .siung *Fuzzy eqv. (3, 9, 3) gram* • Kailan : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Tepung Maizena : 1 .sdt *Fuzzy eqv. (1, 3.13, 1) gram*	• Minyak Goreng : 1 .sdm • Air : 750 .ml • Air : 100 .ml • Minyak Wijen : 1 .sdm • Tepung Maizena : 1 .sdt	
5	Sayur Asem	4	2,1	(500, 1600, 500)	• Labu Siam : 0.666 .buah *Fuzzy eqv. (5, 135.198, 5) gram* • Kacang Panjang : 3 .btg *Fuzzy eqv. (5, 36, 10) gram* • Terong : 1 .buah *Fuzzy eqv. (10, 458, 10) gram* • Kacang Merah : 1 .bks *Fuzzy eqv. (0, 500, 0) gram*	• Air : 2 .gls • Serai : 1 .btg • Daun Melinjo : 10 .lbr • Daun Salam : 1 .lbr • Lada : secukupnya • Garam : secukupnya • Asam Jawa : secukupnya • Bawang Putih : 3 .siung • Bawang Merah : 2 .siung • Cabe Rawit : 2 .buah • Gula : secukupnya	
6	Sayur Bayam Jagung Manis	2	2,1	(500, 2900, 500)	• Bayam : 1 .ikat *Fuzzy eqv. (20, 300, 50) gram* • Jagung Manis : 2 .buah *Fuzzy eqv. (5, 204, 20) gram* • Bawang Putih : 3 .siung *Fuzzy eqv. (3, 9, 3) gram* • Bawang Merah : 2 .siung *Fuzzy eqv. (2, 18, 2) gram*	• Air : 1 .ltr • Garam : 1 .sdt • Gula : secukupnya	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

7	Telur Dadar Jamur	2	2,1,1/2	(500, 2500, 2000)	• Telur : 2 .btr *Fuzzy eqv. (5, 88, 5) gram* • Wortel : 1 .buah *Fuzzy eqv. (5, 61, 5) gram* • Jamur Shiitake : 2 .buah *Fuzzy eqv. (2, 38, 10) gram*	• Minyak Goreng : 2 .sdm • Kaldu Blok : 0.25 .sdt • Keju Cheddar : secukupnya • Garam : 0.5 .sdt • Seledri : secukupnya • Bawang Putih : 3 .siung • Bawang Merah : 2 .siung • Cabe Merah : 1 .buah	
8	Jus Wortel Tomat	1	1,1/2,1/3	(2000, 15100, 500)	• Wortel : 3 .buah *Fuzzy eqv. (5, 183, 5) gram* • Tomat : 3 .buah *Fuzzy eqv. (10, 333, 5) gram*	• Air : 500 .ml	
9	Manggo Smoothies	5	2,1	(500, 4900, 2000)	• Susu Cair : 300 .ml *Fuzzy eqv. (0, 300, 0) gram* • Yogurt Tawar : 3 .sdm *Fuzzy eqv. (2, 45.9, 2) gram* • Air Lemon : 2 .sdm *Fuzzy eqv. (2, 30, 2) gram* • Mangga : 2 .buah *Fuzzy eqv. (5, 672, 5) gram* • Madu : 2 .sdm *Fuzzy eqv. (2, 42, 2) gram*	• Es Batu : secukupnya	
10	Banana Blueberry Smoothies	2	2,1	(500, 3200, 500)	• Susu Cair : 100 .cc *Fuzzy eqv. (0, 100, 0) gram* • Yogurt : 3 .sdm *Fuzzy eqv. (2, 45.9, 2) gram* • Pisang : 1 .buah *Fuzzy eqv. (5, 126, 5) gram* • Blueberry Jam : 1 .sdt	• Es Batu : 4 .buah	

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

				Fuzzy eqv. (1, 5, 2) gram		
11 Ayam Lada Hitam	4	1	(500, 14400, 3000)	• Dada Ayam : 150 .gr *Fuzzy eqv. (0, 150, 0) gram* • Saus Tiram : 3 .sdm *Fuzzy eqv. (3, 54, 3) gram* • Jeruk Nipis : 1 .buah *Fuzzy eqv. (2, 67, 2) gram* • Bawang Putih : 3 .siung *Fuzzy eqv. (2, 9, 2) gram* • Bawang Bombay : 1 .buah *Fuzzy eqv. (10, 331, 2) gram* • Paprika Hijau : 1 .buah *Fuzzy eqv. (5, 119, 5) gram* • Saus Tomat : 2 .sdm *Fuzzy eqv. (2, 32.76, 2) gram* • Paprika Merah : 1 .buah *Fuzzy eqv. (5, 119, 5) gram* • Kecap : 4 .sdm *Fuzzy eqv. (4, 72, 4) gram*	• Lada Hitam : 1 .sdm • Garam : secukupnya • Margarin : secukupnya	
12 Lodeh Rebung	4	1,1/2	(1000, 6500, 500)	• Rebung : 500 .gr *Fuzzy eqv. (0, 500, 0) gram* • Santan : 500 .ml *Fuzzy eqv. (0, 500, 0) gram*	• Penyedap Rasa : secukupnya • Serai : 1 .btg • Laos : 3 .cm • Daun Jeruk : 2 .lbr • Ketumbar : 1 .sdm • Kunyit : 3 .cm • Jinten : 0.5 .sdt • Kemiri : 3 .btr • Daun Salam : 2 .lbr • Garam : secukupnya • Bawang Putih : 3 .siung • Bawang Merah : 7 .siung	

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

						• Cabe Rawit : secukupnya • Cabe Merah : 2 .buah • Gula : secukupnya	
13	Rawon Labu Siam	4	1,1/2	(1000, 8900, 1000)	• Labu Siam : 2 .buah *Fuzzy eqv. (5, 406, 5) gram* • Daging Sapi : 250 .gr *Fuzzy eqv. (0, 250, 0) gram*	• Penyedap Rasa : secukupnya • Air : 2 .ltr • Serai : 1 .btg • Laos : 1 .ruas • Daun Jeruk : 3 .lbr • Ketumbar : 1 .sdt • Kunyit : 3 .cm • Jinten : 0.5 .sdt • Kemiri : 2 .btr • Daun Bawang : secukupnya • Merica : 0.5 .sdt • Daun Salam : 1 .lbr • Garam : secukupnya • Kluwek : 3 .buah • Bawang Putih : 4 .siung • Jahe : 1 .ruas jari • Bawang Merah : 7 .btr • Cabe Merah : 2 .buah • Gula : secukupnya	
14	Perkedel Tahu	8	6,5,4,3,2,1	(100, 500, 200)	• Telur : 1 .btr *Fuzzy eqv. (2, 44, 5) gram* • Tahu : 3 .buah *Fuzzy eqv. (5, 348, 2) gram*	• Merica : secukupnya • Garam : secukupnya • Bawang Putih : 2 .siung • Bawang Merah : 4 .btr	
15	Jus Wortel	1	2,1	(500, 3700, 500)	• Wortel : 2 .buah *Fuzzy eqv. (2, 122, 5) gram* • Gula : 2 .sdm	• Air : 160 .ml	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

					Fuzzy eqv. (1, 25, 1) gram		
16	Jus Mentimun	2	2,1	(100, 600, 200)	• Mentimun : 1 .buah sdg *Fuzzy eqv. (5, 201, 5) gram* • Gula : 2 .sdm *Fuzzy eqv. (2, 25, 2) gram*	• Air : 1.5 .gls blmbg • Es Batu : secukupnya	
17	Jus Alpukat	2	1,1/2	(100, 7200, 500)	• Alpukat : 2 .buah *Fuzzy eqv. (5, 402, 2) gram*	• Air Gula : secukupnya • Air Es : secukupnya	
18	Nasi Goreng Kecap Orak Arik Telor	1	1	(700, 3700, 500)	• Telur : 1 .btr *Fuzzy eqv. (2, 44, 2) gram* • Margarin : 2 .sdm *Fuzzy eqv. (2, 28, 2) gram* • Bawang Putih : 1 .siung *Fuzzy eqv. (1, 3, 2) gram* • Bawang Merah : 1 .siung *Fuzzy eqv. (10, 9, 2) gram* • Nasi : 1 .cup *Fuzzy eqv. (2, 186, 10) gram*	• Cabe Merah : secukupnya • Kecap : secukupnya	
22	Sambal Bawang	10	1,1/2,1/3,1/4	(100, 500, 100)	• Bawang Putih : 8 .siung *Fuzzy eqv. (2, 24, 2) gram* • Cabe Rawit : 0.25 .kg *Fuzzy eqv. (50, 250, 5) gram* • Gula : 1 .sdt *Fuzzy eqv. (1, 4.2, 1) gram*	• Minyak Goreng : secukupnya • Kaldu Bubuk : secukupnya	
23	Sop Iga Sapi	4	1,1/2	(2000, 17200, 2000)	• Buncis : 10 .buah *Fuzzy eqv. (20, 550, 10) gram* • Wortel : 3 .buah *Fuzzy eqv. (5, 183, 5) gram* • Bawang Putih : 3 .siung *Fuzzy eqv. (3, 9, 3) gram*	• Penyedap Rasa : secukupnya • Air : 2 .ltr • Daun Jeruk : 2 .lbr • Daun Bawang : secukupnya	

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

					• Bawang Merah : 8 .siung *Fuzzy eqv. (3, 72, 3) gram* • Kentang : 3 .buah *Fuzzy eqv. (20, 639, 10) gram* • Iga Sapi : 1 .kg *Fuzzy eqv. (0, 1000, 0) gram*	• Lada Bubuk : 1 .sdt • Biji Pala : 1 .buah • Garam : 1.5 .sdm • Seledri : secukupnya • Jahe : 2 .ruas jari	
24	Sup Ayam	1	1,1/2	(2000, 18200, 1000)	• Mentega : 1 .sdt *Fuzzy eqv. (1, 4.73, 1) gram* • Garam : 1 .sdt *Fuzzy eqv. (1, 6, 1) gram* • Dada Ayam : 1 .buah *Fuzzy eqv. (10, 272, 20) gram* • Wortel : 1 .buah *Fuzzy eqv. (2, 61, 2) gram* • Seledri : 1 .tangkai *Fuzzy eqv. (2, 17, 2) gram* • Bawang Putih : 1 .siung *Fuzzy eqv. (1, 3, 1) gram* • Bawang Merah : 2 .siung *Fuzzy eqv. (2, 18, 2) gram* • Kentang : 2 .buah *Fuzzy eqv. (10, 426, 10) gram* • Gula : 0.5 .sdt *Fuzzy eqv. (1, 2.1, 1) gram*	• Air : secukupnya	
25	Sup lobak ala korea	5	1,1/2	(2000, 15300, 1000)	• Minyak Wijen : 2 .sdm *Fuzzy eqv. (2, 27.2, 2) gram* • Wortel : 1 .buah *Fuzzy eqv. (5, 61, 2) gram* • Jagung Manis : 1 .buah *Fuzzy eqv. (2, 102, 10) gram* • Bawang Putih : 1 .sdm *Fuzzy eqv. (1, 5, 1) gram*	• Air : 5 .gls • Daun Bawang : 2 .sdm • Bubuk Cabe Korea : 2 .sdm • Lada : 0.5 .sdt • Garam : secukupnya • Jahe : secukupnya	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

					• Lobak Putih : 1 .buah *Fuzzy eqv. (2, 17, 5) gram* • Daging Sapi : 500 .gr *Fuzzy eqv. (0, 500, 0) gram*		
26	Puding Kukus Pisang Caramel	4	1,1/2,1/4	(2000, 12600, 500)	• Susu Kental Manis : 1 .sachet *Fuzzy eqv. (0, 40, 0) gram* • Telur : 5 .btr *Fuzzy eqv. (10, 220, 10) gram* • Vanili : 0.5 .sdt *Fuzzy eqv. (1, 2.1, 1) gram* • Pisang : 5 .buah *Fuzzy eqv. (20, 630, 10) gram* • Santan : 1 .ltr *Fuzzy eqv. (0, 1000, 0) gram* • Gula : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Gula : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Tepung Maizena : 20 .gr *Fuzzy eqv. (0, 20, 0) gram*	• Air : 20 .ml	
27	Sereal Oat	1	2,1	(500, 1900, 500)	• Sereal, QUAKER Quick Oats : 1 .cup *Fuzzy eqv. (0, 40, 0) gram*		
28	Bolu Panggang	6	6,5,4,3,2,1	(500, 2800, 500)	• Mentega : 300 .gr *Fuzzy eqv. (0, 300, 0) gram* • Susu Kental Manis : 3 .sdm *Fuzzy eqv. (5, 45, 2) gram* • Gula : 250 .gr *Fuzzy eqv. (0, 250, 0) gram* • Terigu : 300 .gr	• Ovalet : 0.5 .sdt	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

					Fuzzy eqv. (0, 300, 0) gram		
29	Susu Kental Manis	1	1	(500, 1600, 500)	• Susu Kental Manis : 1 .sachet *Fuzzy eqv. (0, 40, 0) gram*		
30	Susu	1	1	(1000, 7200, 500)	• Susu Cair : 400 .cc *Fuzzy eqv. (0, 400, 0) gram*		
31	Buah Pisang	1	3,2,1	(200, 1000, 500)	• Pisang : 1 .buah *Fuzzy eqv. (5, 126, 5) gram*		
32	Buah Mangga	1	3,2,1	(1000, 4600, 500)	• Mangga : 1 .buah *Fuzzy eqv. (5, 336, 5) gram*		
33	Buah Jambu Biji	1	3,2,1	(0, 300, 200)	• Jambu Blji : 1 .buah *Fuzzy eqv. (5, 55, 5) gram*		
34	Pecel Bayam Tauge	2	2,1	(1000, 4500, 500)	• Bayam : 1 .ikat *Fuzzy eqv. (10, 300, 20) gram* • Tauge : 0.25 .kg *Fuzzy eqv. (0, 250, 0) gram* • Kacang Tanah : 0.25 .kg *Fuzzy eqv. (0, 250, 0) gram*	• Kencur : 1 .ruas jari • Air : secukupnya • Air : 500 .ml • Daun Jeruk : 1 .lbr • Gula Merah : 2 .biji • Terasi : 0.25 .sdt • Bawang Putih : 3 .siung • Bawang Merah : 3 .siung • Cabe Rawit : 5 .biji • Cabe Merah : 5 .biji	
35	Buah Pepaya	1	3,2,1	(300, 800, 200)	• Pepaya : 0.25 .buah bsr *Fuzzy eqv. (0, 195.25, 0) gram*		
36	Bubur Ayam	3	1,1/2	(500, 4600,	• Telur : 2 .btr	• Minyak Goreng :	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

				500)	*Fuzzy eqv. (2, 88, 2) gram* • Dada Ayam : 1 .buah *Fuzzy eqv. (5, 272, 5) gram* • Beras Ketan Putih : 50 .gr *Fuzzy eqv. (0, 50, 0) gram* • Beras : 150 .gr *Fuzzy eqv. (0, 150, 0) gram*	secukupnya • Air : 500 .ml • Daun Bawang : 3 .btg • Cakue : 2 .buah	
37	Sayur Bening Bayam	1	1,1/2	(1000, 6000, 500)	• Bayam : 1 .ikat *Fuzzy eqv. (10, 300, 10) gram* • Jagung Manis : 1 .buah *Fuzzy eqv. (5, 102, 10) gram* • Bawang Merah : 2 .siung *Fuzzy eqv. (2, 18, 2) gram* • Kacang Merah : 1 .genggam *Fuzzy eqv. (5, 197, 10) gram*	• Temu Kunci : 1 .ruas • Bawang Goreng : secukupnya • Garam : secukupnya • Gula : secukupnya	
38	Roti Tawar Selai Blueberry	1	2,1,1/2	(500, 5900, 500)	• Blueberry Jam : 2 .sdm *Fuzzy eqv. (2, 21.2, 5) gram* • Roti Tawar : 2 .lbr *Fuzzy eqv. (5, 56, 5) gram*		
39	Capcay	4	1,1/2,1/4	(1000, 10300, 2000)	• Sosis Sapi : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Buncis : 100 .gr *Fuzzy eqv. (0, 100, 0) gram* • Brokoli : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Wortel : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Bawang Putih : 3 .siung *Fuzzy eqv. (3, 9, 3) gram*	• Minyak Goreng : secukupnya • Air : 100 .ml • Daun Bawang : 2 .btg • Kaldu Bubuk : 1 .sdm • Biji Pala : 0.25 .buah • Lada : 0.5 .sdt	

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

					• Bawang Merah : 2 .siung *Fuzzy eqv. (2, 18, 2) gram* • Udang : 250 .gr *Fuzzy eqv. (0, 250, 0) gram* • Gula : 0.25 .sdt *Fuzzy eqv. (0.5, 1.05, 0.5) gram*		
40	Omelet Kentang	4	2,1,1/2	(500, 2300, 1000)	• Telur : 2 .buah *Fuzzy eqv. (2, 88, 2) gram* • Wortel : 1 .buah *Fuzzy eqv. (2, 61, 2) gram* • Seledri : 1 .helai *Fuzzy eqv. (5, 17, 1) gram* • Bawang Putih : 1 .btr *Fuzzy eqv. (1, 3, 1) gram* • Kentang : 2 .buah sdg *Fuzzy eqv. (5, 426, 5) gram*	• Merica Bubuk : 0.5 .sdt • Garam : 0.5 .sdt	
41	Buah Apel	1	2,1	(1000, 4000, 500)	• Apel : 1 .buah *Fuzzy eqv. (2, 161, 5) gram*		
42	Brownis Pisang	4	4,3,2,1	(500, 5000, 1000)	• Telur : 2 .btr *Fuzzy eqv. (5, 88, 5) gram* • Margarin : 3 .sdm *Fuzzy eqv. (3, 42, 3) gram* • Pisang : 4 .buah *Fuzzy eqv. (10, 504, 5) gram* • Coklat Bubuk : 2 .sdm *Fuzzy eqv. (5, 90, 5) gram* • Coklat Batang : 100 .gr *Fuzzy eqv. (0, 100, 0) gram* • Gula : 100 .gr *Fuzzy eqv. (0, 100, 0) gram* • Terigu : 100 .gr	• Minyak Goreng : 2 .sdm • Ovalet : 1 .sdt • Baking Powder : 0.5 .sdt	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

					Fuzzy eqv. (0, 100, 0) gram		
43	Donat Kentang	15	4,3,2,1	(500, 1200, 500)	• Mentega : 75 .gr *Fuzzy eqv. (0, 75, 0) gram* • Susu Bubuk : 50 .gr *Fuzzy eqv. (0, 50, 0) gram* • Telur : 2 .btr *Fuzzy eqv. (5, 88, 5) gram* • Garam : 0.25 .sdt *Fuzzy eqv. (0.5, 1.5, 0.5) gram* • Kentang : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* • Gula : 100 .gr *Fuzzy eqv. (0, 100, 0) gram* • Terigu Protein Tinggi : 500 .gr *Fuzzy eqv. (0, 500, 0) gram*	• Air Es : 100 .ml • Ragi : 11 .gr	
44	Soto Mie Bogor	5	1,1/2,1/4	(2000, 17100, 500)	• Kentang : 2 .buah bsr *Fuzzy eqv. (10, 738, 5) gram* • Tomat : 1 .buah *Fuzzy eqv. (5, 111, 5) gram* • Daging Sapi : 500 .gr *Fuzzy eqv. (0, 500, 0) gram* • Mie Telor : 1 .bks *Fuzzy eqv. (0, 200, 0) gram*	• Lengkuas : 1 .ruas • Air : 2 .ltr • Serai : 1 .tangkai • Daun Jeruk : 2 .lbr • Kemiri : 5 .btr • Daun Bawang : 2 .tangkai • Merica : secukupnya • Bawang Goreng : secukupnya • Risol Bihun : secukupnya • Daun Salam : 2 .lbr • Garam : secukupnya • Kol : secukupnya • Bawang Putih : 5 .siung • Jahe : 1 .ruas • Bawang Merah : 10 .siung	

UNIVERSITAS BRAWIJAYA
repository.ub.ac.id

						• Cabe Merah : 3 .buah • Gula : secukupnya	
45	Bihun Tumis	5	1	(1000, 3700, 500)	• Seledri : 1 .btg *Fuzzy eqv. (2, 17, 2) gram* • Bawang Putih : 3 .buah *Fuzzy eqv. (3, 9, 3) gram* • Bawang Merah : 5 .buah *Fuzzy eqv. (5, 45, 5) gram* • Tomat : 2 .buah *Fuzzy eqv. (5, 222, 5) gram* • Kecap : 2 .sdm *Fuzzy eqv. (2, 36, 2) gram* • Bihun : 200 .gr *Fuzzy eqv. (0, 200, 0) gram* •	• Minyak Goreng : 3 .sdm • Air : secukupnya • Daun Bawang : 1 .btg • Merica : 0.5 .sdt • Garam : 0.5 .sdt	
46	Kue Lumpur Sari Pandan	10	4,3,2,1	(1000, 3000, 500)	• Telur : 2 .btr *Fuzzy eqv. (5, 88, 5) gram* • Kentang : 5 .buah *Fuzzy eqv. (5, 1065, 10) gram* • Santan : 350 .ml *Fuzzy eqv. (0, 350, 0) gram* • Gula : 5 .sdm *Fuzzy eqv. (5, 62.5, 5) gram* • Terigu : 300 .gr *Fuzzy eqv. (0, 300, 0) gram*	• Sari Daun Pandan : 3 .sdm • Vanili : secukupnya	
47	Sari Tomat Segar	4	2,1	(1000, 4900, 500)	• Tomat : 6 .buah *Fuzzy eqv. (10, 666, 5) gram*	• Air : 1.5 .ltr • Daun Pandan : 1 .lbr • Gula : secukupnya	
48	Bubur Kacang Hijau	6	2,1,1/2,1/4	(1000, 6100, 1000)	• Garam : 0.5 .sdt *Fuzzy eqv. (1, 3, 1) gram*	• Pacar Cina : 100 .gr • Daun Pandan : 2 .lbr	

UNIVERSITAS BRAWIJAYA repository.ub.ac.id

				• Santan : 1 .ltr *Fuzzy eqv. (0, 1000, 0) gram* • Kacang Hijau : 250 .gr *Fuzzy eqv. (0, 250, 0) gram* • Gula : 50 .gr *Fuzzy eqv. (0, 50, 0) gram*	• Gula Jawa : 200 .gr	

LAMPIRAN L DATA MENU SET

Data Menu Set

Id	Nama	Menu	Valid Sebagai Data Uji
1	Menu Set 01	• Sambal Goreng Ampela • Nasi • Oseng-oseng Kacang Panjang Campur Tempe • Nasi • Brownis Pisang • Jus Wortel Tomat	
2	Menu Set 02	• Nasi Goreng Kecap Orak Arik Telor • Susu Kental Manis • Nasi • Sayur Asem	
3	Menu Set 03	• Nasi • Telur Dadar Jamur • Susu Kental Manis • Lodeh Rebung • Nasi • Ayam Lada Hitam • Nasi	
4	Menu Set 04	• Susu • Rawon Labu Siam • Nasi • Jus Wortel • Bolu Panggang	
5	Menu Set 05	• Manggo Smoothies • Bolu Panggang • Nasi • Sayur Asem • Nasi Goreng Kecap Orak Arik Telor • Susu Kental Manis	
6	Menu Set 06	• Susu Kental Manis • Nasi • Sayur Bayam Jagung Manis • Jus Wortel Tomat • Buah Pepaya • Donat Kentang • Susu Kental Manis	
7	Menu Set 07	• Sereal Oat • Buah Pisang • Nasi • Lodeh Rebung • Nasi • Sambal Bawang	

		• Perkedel Tahu	
8	Menu Set 08	• Nasi • Pecel Bayam Tauge • Buah Pisang • Nasi • Lodeh Rebung • Bihun Tumis • Jus Wortel Tomat	
9	Menu Set 09	• Nasi • Pecel Bayam Tauge • Buah Pisang • Nasi • Sayur Asem • Nasi • Perkedel Tahu • Oseng-oseng Kacang Panjang Campur Tempe	
10	Menu Set 10	• Bubur Ayam • Nasi • Sayur Bening Bayam • Nasi • Sup Ayam • Buah Pepaya	
11	Menu Set 11	• Susu Kental Manis • Roti Tawar Selai Blueberry • Nasi • Pecel Bayam Tauge • Capcay • Nasi • Ayam Lada Hitam • Buah Pepaya • Donat Kentang	
12	Menu Set 12	• Sereal Oat • Susu • Nasi • Cah Kailan • Nasi Goreng Kecap Orak Arik Telor	
13	Menu Set 13	• Omelet Kentang • Nasi • Rawon Labu Siam • Nasi • Telur Dadar Jamur	
14	Menu Set 14	• Susu Kental Manis • Nasi Goreng Kecap Orak Arik Telor	

		• Bolu Panggang • Nasi • Capcay • Telur Dadar Jamur • Buah Pepaya	
15	Menu Set 15	• Telur Dadar Jamur • Nasi • Buah Pepaya • Brownis Pisang	
16	Menu Set 16	• Buah Apel • Sereal Oat • Susu • Nasi • Pecel Bayam Tauge • Jus Alpukat • Nasi • Sambal Goreng Ampela	
17	Menu Set 17	• Jus Wortel • Bubur Ayam • Nasi • Sup lobak ala korea • Nasi • Capcay • Bubur Kacang Hijau • Buah Pepaya • Sayur Bening Bayam • Sereal Oat	
18	Menu Set 18	• Nasi • Perkedel Tahu • Sayur Bening Bayam • Sambal Bawang • Nasi • Buah Pisang • Nasi Goreng Kecap Orak Arik Telor • Donat Kentang	
19	Menu Set 19	• Nasi • Sayur Bayam Jagung Manis • Susu • Bolu Panggang • Nasi • Sayur Bening Bayam • Buah Pepaya	
20	Menu Set 20	• Susu Kental Manis • Nasi • Sayur Asem	

		• Nasi • Jus Mentimun • Buah Mangga • Brownis Pisang	
21	Menu Set 21	• Pecel Bayam Tauge • Nasi • Buah Pisang • Capcay • Nasi	
22	Menu Set 22	• Pecel Bayam Tauge • Nasi • Buah Mangga • Donat Kentang • Nasi • Cah Kailan • Banana Blueberry Smoothies	
23	Menu Set 23	• Buah Apel • Omelet Kentang • Buah Pepaya • Brownis Pisang • Jus Alpukat	
24	Menu Set 24	• Jus Mentimun • Jus Alpukat • Jus Wortel Tomat • Nasi • Telur Dadar Jamur	
25	Menu Set 25	• Brownis Pisang • Roti Tawar Selai Blueberry • Sereal Oat • Susu • Nasi • Sayur Bayam Jagung Manis • Brownis Pisang • Susu Kental Manis	
26	Menu Set 26	• Jus Wortel Tomat • Nasi • Telur Dadar Jamur • Sup lobak ala korea • Nasi • Jus Mentimun	
27	Menu Set 27	• Pecel Bayam Tauge • Nasi • Donat Kentang • Buah Pisang	

		• Sayur Bening Bayam • Nasi	
28	Menu Set 28	• Bubur Ayam • Buah Pisang • Nasi • Sop Iga Sapi • Susu	
29	Menu Set 29	• Puding Kukus Pisang Caramel • Nasi • Cah Kailan • Nasi • Capcay • Jus Alpukat	
30	Menu Set 30	• Omelet Kentang • Nasi • Sup lobak ala korea • Nasi • Rawon Labu Siam	
31	Menu Set 31	• Nasi Goreng Kecap Orak Arik Telor • Susu Kental Manis • Nasi • Pecel Bayam Tauge • Buah Mangga • Sayur Bening Bayam • Nasi • Perkedel Tahu	
32	Menu Set 32	• Soto Mie Bogor • Nasi • Rawon Labu Siam • Nasi • Perkedel Tahu	
33	Menu Set 33	• Susu • Bolu Panggang • Nasi • Sup lobak ala korea • Buah Pisang • Buah Pepaya • Sayur Bening Bayam	
34	Menu Set 34	• Sambal Goreng Ampela • Nasi • Nasi • Sayur Bening Bayam • Nasi • Buah Pepaya	

35	Menu Set 35	• Roti Tawar Selai Blueberry • Buah Pisang • Susu • Lodeh Rebung • Nasi • Nasi • Omelet Kentang • Donat Kentang • Susu	
36	Menu Set 36	• Puding Kukus Pisang Caramel • Susu Kental Manis • Sayur Bening Bayam • Buah Pepaya • Nasi • Nasi Goreng Kecap Orak Arik Telor	
37	Menu Set 37	• Brownis Pisang • Nasi • Perkedel Tahu • Sambal Bawang • Nasi • Buah Pepaya	
38	Menu Set 38	• Donat Kentang • Sereal Oat • Nasi • Omelet Kentang • Jus Alpukat	
39	Menu Set 39	• Nasi • Sop Iga Sapi • Bubur Ayam • Brownis Pisang	
40	Menu Set 40	• Susu Kental Manis • Nasi • Rawon Labu Siam • Buah Jambu Biji • Bubur Ayam	
41	Menu Set 41	• Kue Lumpur Sari Pandan • Susu • Lodeh Rebung • Nasi • Buah Pisang • Nasi • Sayur Bening Bayam • Buah Pepaya	

42	Menu Set 42	• Nasi • Perkedel Tahu • Sambal Bawang • Capcay • Cah Kailan • Donat Kentang	
43	Menu Set 43	• Bihun Tumis • Susu • Nasi • Sayur Bayam Jagung Manis • Buah Jambu Biji • Nasi • Sup lobak ala korea	
44	Menu Set 44	• Sari Tomat Segar • Nasi • Sayur Bayam Jagung Manis • Nasi • Perkedel Tahu • Cah Kailan • Nasi • Brownis Pisang	
45	Menu Set 45	• Brownis Pisang • Buah Pepaya	
46	Menu Set 46	• Nasi • Sup lobak ala korea • Bubur Ayam • Jus Wortel Tomat • Sayur Asem • Nasi • Perkedel Tahu • Susu	
47	Menu Set 47	• Nasi • Banana Blueberry Smoothies • Brownis Pisang • Nasi • Sayur Bayam Jagung Manis • Buah Pepaya • Perkedel Tahu	
48	Menu Set 48	• Banana Blueberry Smoothies • Roti Tawar Selai Blueberry • Nasi • Sup Ayam • Buah Jambu Biji • Kue Lumpur Sari Pandan • Nasi • Perkedel Tahu	

		• Cah Kailan • Susu	
49	Menu Set 49	• Sop Iga Sapi • Nasi • Jus Wortel • Buah Mangga • Oseng-oseng Kacang Panjang Campur Tempe • Nasi • Bihun Tumis • Kue Lumpur Sari Pandan	
50	Menu Set 50	• Susu • Buah Pisang • Nasi • Sup Ayam • Nasi • Ayam Lada Hitam • Banana Blueberry Smoothies	