

OPTIMASI BIAYA PEMENUHAN GIZI DAN NUTRISI PADA ORANG DEWASA MENGGUNAKAN METODE SIMPLEKS BIG-M

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Akbar Hakim Prabowo
NIM: 115060800111034



INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

OPTIMASI BIAYA PEMENUHAN GIZI DAN NUTRISI PADA ORANG DEWASA
MENGUNAKAN METODE SIMPLEKS BIG-M

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Akbar Hakim prabowo
NIM: 115060800111034

Skripsi ini telah diuji dan dinyatakan lulus pada
14 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Drs. Marji, M.T.
NIP: 19670801 1999203 1 001

Edy Santoso, S.Si., M.Kom
NIP: 19740414 200312 1 004

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP: 19670801 1999203 1 001

PERNYATAAN ORISINALITAS

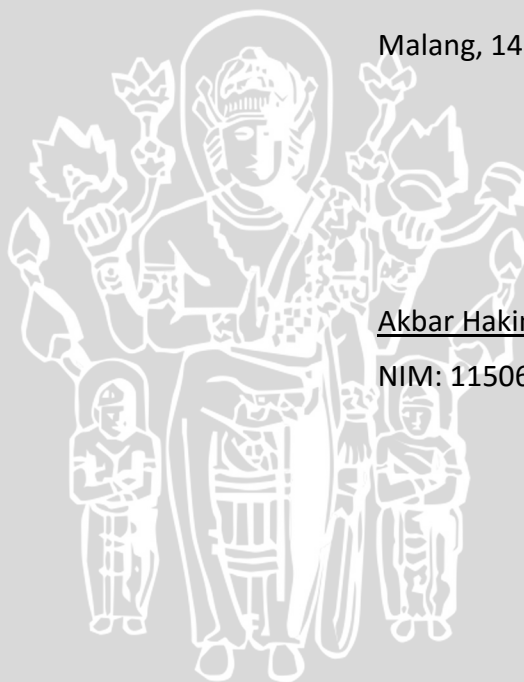
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan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

Apabila ternyata didalam 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, 14 Januari 2016

Akbar Hakim Prabowo

NIM: 115060800111034



KATA PENGANTAR

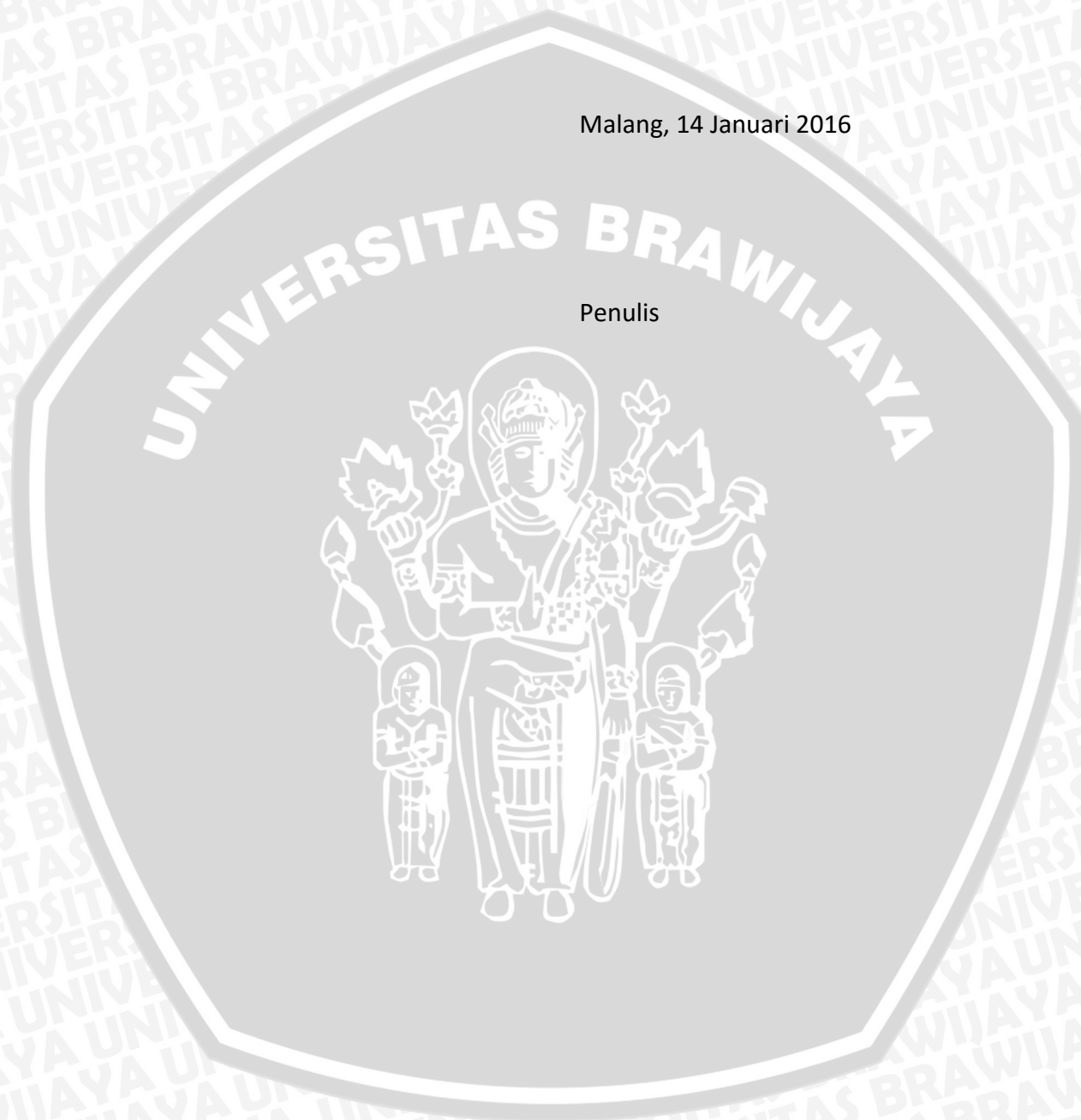
Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Penyayang. Puji syukur penulis panjatkan kehadirat Allah SWT karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul **“Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Orang Dewasa Menggunakan Metode Simpleks-Big-M”**. Shalawat dan salam atas junjungan Nabi besar kita Nabi Muhammad SAW beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang. Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi, diantaranya:

1. Drs. Marji, M.T., dan Edy Santoso, S.Si, M.Kom., selaku dosen pembimbing yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini .
2. Orang tua penulis, Bapak Sugiyono dan Ibu Endang Mulyaningsih beserta keluarga yang telah memberikan dukungan moral maupun material.
3. Drs. Marji, M.T., dan Issa Arwani, S.Kom, M.Sc., selaku Ketua dan Sekretaris Program Studi Informatika/Illmu Komputer Universitas Brawijaya.
4. Seluruh Civitas Akademika Informatika/Illmu Komputer Universitas Brawijaya yang telah banyak memberi dukungan dan bantuan selama penulis menempuh studi di Informatika/Illmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Seluruh dosen Informatika/Illmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Sahabat Penulis, rekan seperjuangan sejak semester awal Informatika 2011 Kelas F (SCANF), rekan seperjuangan pengerjaan skripsi (Cusmonout), rekan panitia probinmaba 2013 divisi acara, rekan Informatika 2009, 2010, 2011, 2012 dan 2013 pada umumnya. Yang selalu memberikan dukungan, motivasi, kritik dan saran.
7. Seluruh anggota HMIF PTIIK UB, pengurus EMIF dan BPMIF, LKM PTIIK UB pada khususnya dan KBMTIIK pada umumnya. Yang tidak hanya memberikan dukungan, motivasi, kritik maupun saran namun memberikan wadah dan membentuk penulis untuk mengembangkan diri menjadi pribadi yang lebih baik.

Hanya doa yang penulis bisa berikan semoga Allah SWT memberikan balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat khususnya bagi diri sendiri dan bagi semua pihak.

Malang, 14 Januari 2016

Penulis



ABSTRAK

Seiring dengan semakin meningkatnya harga bahan-bahan pangan, masyarakat dituntut untuk semakin pandai dalam menyiasati menu harian mereka. Namun karena batasan biaya, sering kali kandungan gizi pada menu harian dikesampingkan. Untuk mengatasi masalah pemenuhan gizi ini, diperlukan suatu langkah optimasi. Tujuan optimasi ini ialah untuk mendapatkan asupan gizi yang sesuai kebutuhan serta biaya yang minimum. Dalam studi kasus ini digunakan rumus Harris Benedict untuk menghitung tingkat kebutuhan individu. Metode Simpleks Big-M digunakan untuk mendapatkan hasil berupa kuantitas makanan yang dipilih beserta total harga yang paling optimal. Solusi optimal untuk nilai minimum konsumsi adalah 0,1 satuan atau setara 15 gram. Selain itu aplikasi dapat menentukan kuantitas makanan yang dikonsumsi dan jumlah biaya yang harus dikeluarkan secara optimal.

Kata Kunci : Nutrisi, Sistem Pendukung Keputusan, Metode Simpleks Big-M



ABSTRACT

Along with the increasing prices of food, people are required to deal with increasingly clever in their daily menu. However, due to cost constraints, often the nutrients in the daily menu ruled out. To overcome the problem of the fulfillment of this nutrient, required an optimization step. The goal of this optimization is to get appropriate nutrition needs and minimum cost. In this case study is used Harris Benedict formula to calculate the level of individual needs. Simplex Big-M method used to obtain the results in the form of the quantity of food that is selected along with the total price of the most optimal. The optimal solution for the minimum value of consumption is 0.1 units or the equivalent of 15 grams. Further, the application can determine the quantity of food consumed and the amount of fees that must be spent optimally.

Keywords : Nutrition, Decision Support System, Simplex Big-M Method



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR SOURCE CODE	xv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian pustaka	5
2.2 Diet problem	5
2.3 DRI dan AKG	6
2.4 Harris benedict equation	8
2.5 Pemrograman linear.....	9
2.6 Metode Simpleks	10
2.6.1 Variabel basis dan nonbasis	13
2.6.2 Metode Big-M	13
2.6.3 Penerapan metode Simpleks Big-M.....	13
BAB 3 METODOLOGI	15
3.1 Tahapan penelitian	15
3.2 Kebutuhan sistem	16

3.2.1 Deskripsi umum sistem	16
3.2.2 Data yang digunakan	17
3.3 Formulasi permasalahan	17
BAB 4 PERANCANGAN	18
4.1 Perancangan algoritma	18
4.1.1 Merumuskan kebutuhan user	19
4.1.2 Memasukkan data makanan	20
4.1.3 Membuat fungsi tujuan	21
4.1.4 Membuat fungsi kendala	22
4.1.5 Ubah model standar	23
4.1.6 Proses iterasi	25
4.2 Perhitungan manual	33
4.2.1 Menentukan kebutuhan nutrisi dan menu makanan	33
4.2.2 Mengubah ke bentuk <i>pemrograman linear</i>	34
4.2.3 Mengubah ke model standar	35
4.2.4 Mengubah ke tabel matriks	36
4.3 Perancangan database	46
4.4 Perancangan antarmuka	48
4.5 Perancangan ujicoba dan evaluasi	50
4.5.1 Pengujian validitas hasil perhitungan	50
4.5.2 Pengujian sensitivitas nilai minimum konsumsi makanan	50
4.5.3 Pengujian sensitivitas jenis dan jumlah pilihan makanan	51
BAB 5 IMPLEMENTASI	52
5.1 Lingkungan implementasi	52
5.1.1 Lingkungan perangkat keras	52
5.1.2 Lingkungan perangkat lunak	52
5.2 Implementasi program	52
5.2.1 Implementasi proses input data individu	53
5.2.2 Implementasi proses pilih makanan	54
5.2.3 Implementasi proses dapat makanan	55
5.2.4 Implementasi proses menghitung nutrisi individu	55
5.2.5 Implementasi proses fungsi tujuan	57

5.2.6 Implementasi proses fungsi kendala	57
5.2.7 Implementasi proses model standar	59
5.2.8 Implementasi proses membuat tabel Simpleks Big-M awal.....	60
5.2.9 Implementasi proses menghitung Zj dan Cj-Zj.....	65
5.2.10 Implementasi proses menentukan kolom kunci.....	65
5.2.11 Implementasi proses menentukan baris kunci	66
5.2.12 Implementasi proses revisi koefisien baru	67
5.2.13 Implementasi proses cek positif nilai CjZj.....	68
5.2.14 Implementasi proses buat result	68
5.2.15 Implementasi proses main program.....	70
5.3 Implementasi antarmuka.....	72
5.3.1 Halaman index.....	72
5.3.2 Halaman sign up.....	74
5.3.3 Halaman home.....	75
5.3.4 Halaman profil.....	75
5.3.5 Halaman rekam konsumsi.....	76
5.3.6 Halaman menu makanan	76
5.3.7 Halaman pilih makanan.....	77
5.3.8 Halaman nutrisipuser	77
BAB 6 PENGUJIAN DAN ANALISIS.....	80
6.1 Sistematika pengujian.....	80
6.1.1 Sistematika pengujian validitas hasil perhitungan.....	80
6.1.2 Sistematika pengujian sensitivitas nilai minimal konsumsi makanan.....	80
6.1.3 Sistematika pengujian sensitivitas jenis dan jumlah pilihan makanan.....	80
6.2 Analisa dan hasil pembahasan.....	80
6.2.1 Hasil pengujian validitas hasil perhitungan	80
6.2.2 Hasil pengujian sensitivitas nilai minimal konsumsi makanan ...	83
6.2.3 Hasil pengujian sensitivitas jenis dan jumlah pilihan makanan..	84
BAB 7 Penutup	87
7.1 Kesimpulan.....	87

7.2 Saran	87
DAFTAR PUSTAKA.....	88
LAMPIRAN DATA DAFTAR MAKANAN.....	89



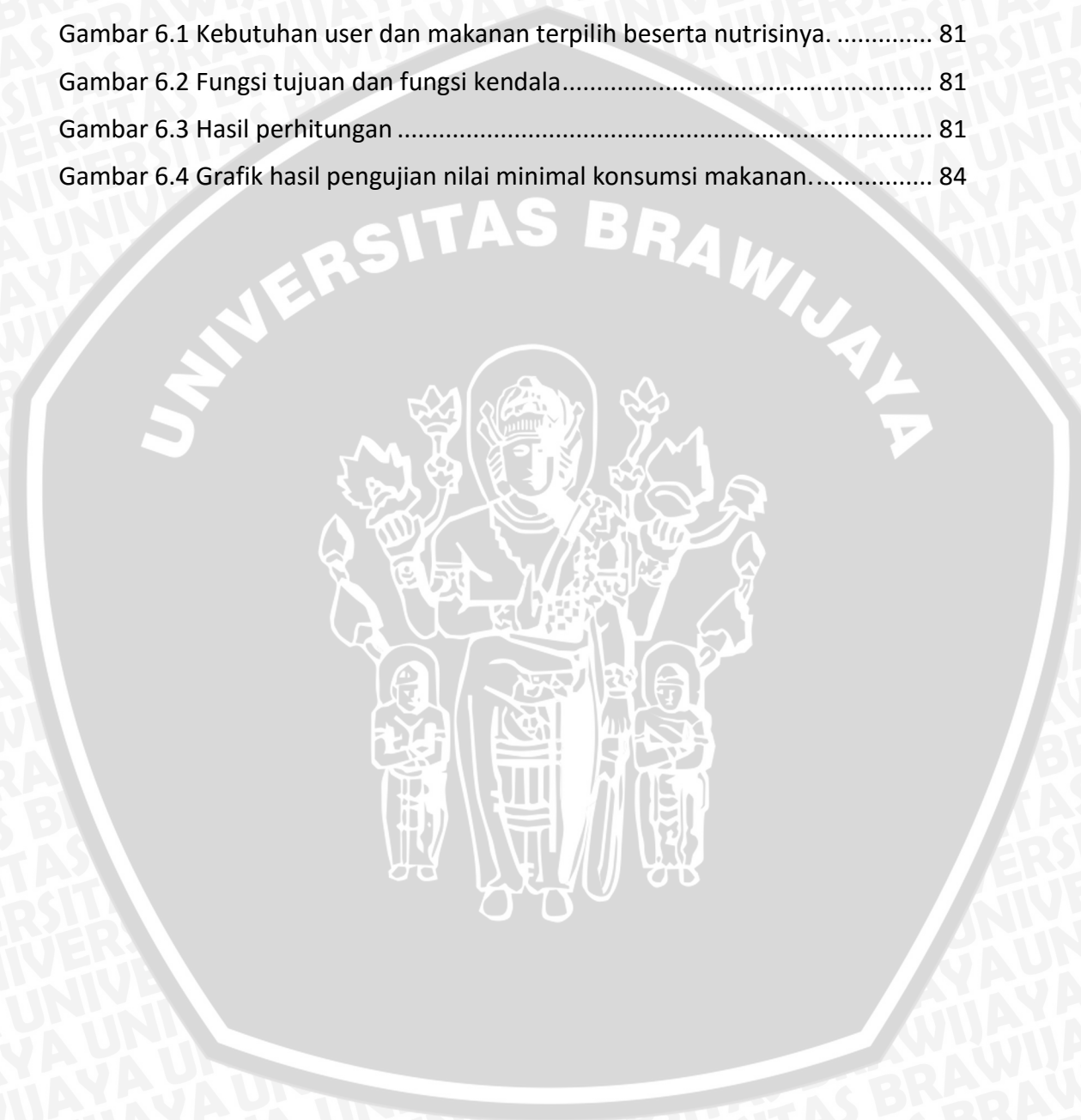
DAFTAR TABEL

Tabel 2.1 Tingkat aktivitas TEE	8
Tabel 2.2 Variabel <i>Slack</i>	11
Tabel 2.3 Variabel <i>Surplus</i>	12
Tabel 4.1 Tabel makanan beserta nutrisi dan harga	34
Tabel 4.2 Tabel awal Simpleks	36
Tabel 4.3 Tabel menentukan variabel yang akan diganti	37
Tabel 4.4 Menghitung koefisien baru untuk variabel yang masuk	38
Tabel 4.5 Revisi baris lain	39
Tabel 4.6 Tabel Simpleks II	40
Tabel 4.7 Tabel Simpleks III	41
Tabel 4.8 Tabel Simpleks IV	41
Tabel 4.9 Tabel Simpleks V	42
Tabel 4.10 Tabel Simpleks VI	42
Tabel 4.11 Tabel Simpleks VII	43
Tabel 4.12 Tabel Simpleks VIII	43
Tabel 4.13 Tabel Simpleks IX	44
Tabel 4.14 Atribut user	46
Tabel 4.15 Atribut profil_individu	47
Tabel 4.16 Atribut aktifitas	47
Tabel 4.17 Atribut makanan	47
Tabel 4.18 Atribut rekam_kebutuhan	47
Tabel 4.19 Rancangan pengujian sensitivitas nilai minimum konsumsi makanan	51
Tabel 4.20 Rancangan pengujian sensitivitas jumlah pilihan makanan	51
Tabel 6.1 Hasil pengujian sensitivitas nilai minimal konsumsi makanan	83
Tabel 6.2 Hasil uji 1 – makanan dipilih secara random	84
Tabel 6.3 Hasil uji 2 – makanan dipilih berdasarkan harga terendah	85
Tabel 6.4 Hasil uji 3 – makanan dipilih berdasarkan harga tertinggi	85
Tabel 6.5 Keterangan indeks pengujian	85

DAFTAR GAMBAR

Gambar 1.1 Kecenderungan status gizi penduduk dewasa (diatas 18 tahun) berdasarkan komposit TB dan IMT Indonesia 2010 – 2013	2
Gambar 2.1 hubungan RDA,AI dan UL	7
Gambar 3.1 Tahap penelitian.....	15
Gambar 4.1 Diagram alir sistem.....	18
Gambar 4.2 <i>Flowchart</i> proses optimalisasi biaya dan menu	19
Gambar 4.3 <i>Flowchart</i> merumuskan kebutuhan user.	20
Gambar 4.4 <i>Flowchart</i> memasukkan data makanan.	21
Gambar 4.5 <i>Flowchart</i> membuat fungsi tujuan.....	22
Gambar 4.6 <i>Flowchart</i> membuat fungsi kendala.....	23
Gambar 4.7 <i>Flowchart</i> ubah ke model standar.	24
Gambar 4.8 <i>Flowchart</i> proses iterasi.	25
Gambar 4.9 <i>Flowchart</i> membuat tabel matriks.....	27
Gambar 4.10 <i>Flowchart</i> menghitung Z_j dan $C_j - Z_j$	28
Gambar 4.11 <i>Flowchart</i> pilih kolom negatif terbesar $C_j - Z_j$	29
Gambar 4.12 <i>Flowchart</i> pilih baris yang akan diganti.....	30
Gambar 4.13 <i>Flowchart</i> koefisien baru variabel masuk.	32
Gambar 4.14 Desain database	46
Gambar 4.15 Perancangan antarmuka input biodata individu	48
Gambar 4.16 Perancangan antarmuka input pilihan makanan	49
Gambar 4.17 Perancangan antarmuka output matriks makanan	50
Gambar 5.1 Tampilan halaman <i>index (menu)</i>	73
Gambar 5.2 Tampilan halaman <i>index (about)</i>	73
Gambar 5.3 Tampilan halaman <i>index (sign in)</i>	74
Gambar 5.4 Tampilan halaman <i>index (contact)</i>	74
Gambar 5.5 Tampilan halaman <i>sign up</i>	75
Gambar 5.6 Tampilan halaman <i>home</i>	75
Gambar 5.7 Tampilan halaman profil.	76
Gambar 5.8 Tampilan rekam konsumsi	76
Gambar 5.9 Tampilan halaman menu makanan.....	77

Gambar 5.10 Tampilan halaman pilih makanan.	77
Gambar 5.11 Tampilan halaman nutriuser (pemrograman linear).....	78
Gambar 5.12 Tampilan halaman nutriuser (tabel matriks).	79
Gambar 5.13 Tampilan halaman nutriuser (tabel result).	79
Gambar 6.1 Kebutuhan user dan makanan terpilih beserta nutrisinya.	81
Gambar 6.2 Fungsi tujuan dan fungsi kendala.....	81
Gambar 6.3 Hasil perhitungan	81
Gambar 6.4 Grafik hasil pengujian nilai minimal konsumsi makanan.....	84



DAFTAR SOURCE CODE

Soucecode 5.1 Model input data individu	53
Soucecode 5.2 Controller input data individu	54
Soucecode 5.3 Model pilih makanan	54
Soucecode 5.4 Controller pilih makanan	55
Soucecode 5.5 Model dapat makanan.....	55
Soucecode 5.6 Contoller dapat makanan	55
Soucecode 5.7 Controller menghitung nutrisi individu	57
Soucecode 5.8 Controller fungsi tujuan.....	57
Soucecode 5.9 Controller fungsi kendala	59
Soucecode 5.10 Controller model standar	60
Soucecode 5.11 Controller membuat tabel awal	65
Soucecode 5.12 Controller menghitung Z_j dan $C_j - Z_j$	65
Soucecode 5.13 Controller menentukan kolom kunci.....	66
Soucecode 5.14 Controller menentukan baris kunci.....	67
Soucecode 5.15 Controller revisi koefisien baru	68
Soucecode 5.16 Controller cek positif nilai $C_j Z_j$	68
Soucecode 5.17 Controller menampilkan result	70
Soucecode 5.18 Contoller main program	72

DAFTAR LAMPIRAN

Lampiran 1 Data daftar makanan satuan/150 gr (Depkes 2010 & Penelitian (Suci, 2015)) 89



BAB 1 PENDAHULUAN

1.1 Latar belakang

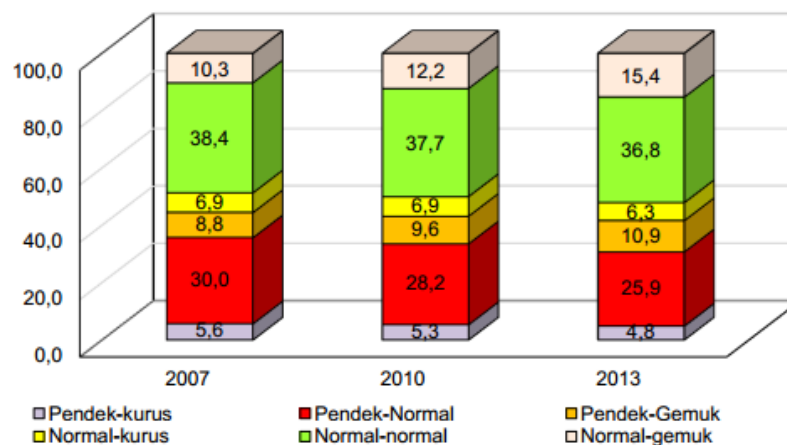
Seiring dengan semakin meningkatnya harga bahan-bahan pangan, masyarakat dituntut untuk semakin pandai dalam menyiasati menu harian mereka. Namun karena batasan biaya, sering kali kandungan gizi pada menu harian dikesampingkan. Padahal, dengan melakukan variasi pada menu akan sangat mungkin untuk mendapatkan asupan gizi yang cukup dengan biaya terjangkau. Untuk mengatasi masalah pemenuhan gizi ini, diperlukan suatu langkah optimasi. Tujuan optimasi ini ialah untuk mencari komposisi menu optimal, dimana asupan gizinya mencukupi serta biaya yang diperlukan masih dapat dijangkau.

Permasalahan pemenuhan kebutuhan gizi atau dikenal pula dengan diet problem merupakan salah satu permasalahan klasik yang mulai dipelajari pada tahun 1940-an. Berawal dari pemenuhan diet sehat bagi tentara Amerika, diet problem berkembang hingga saat ini. Menurut Matthew Ford, tujuan utama pemecahan diet problem adalah untuk menemukan kombinasi makanan termurah yang dapat memenuhi kebutuhan nutrisi harian minimal seseorang.

Di Indonesia sendiri dalam lima tahun terakhir ini pembangunan kesehatan telah diperkuat dengan tersedianya data dan informasi yang dihasilkan oleh Riset Kesehatan Dasar atau Riskesdas. Tiga Riskesdas telah dilaksanakan di Indonesia, masing-masing pada tahun 2007, 2010, dan 2013. Riset Kesehatan Dasar (Riskesdas) 2013 merupakan riset kedua yang mengumpulkan data dasar dan indikator kesehatan setelah tahun 2007 yang merepresentasikan gambaran wilayah nasional, provinsi, dan kabupaten/kota. Populasi dalam Riskesdas 2013 adalah seluruh rumah tangga di 33 provinsi, 497 kabupaten/kota.

Salah satu Riset yang dilakukan dalam Riskesdas tahun 2013 adalah mengenai survei status gizi. Survei yang berkenaan dalam kategori status gizi penduduk dewasa terdapat sejumlah 649625 responden dimana didalamnya juga terdiri status gizi wanita usia subur sejumlah 272556 dan ibu hamil sejumlah 7664. Prevalensi penduduk dewasa Indonesia pada tahun 2013 mendapati angka untuk kurus 8,7 persen, berat badan lebih 13,5 persen dan obesitas 15,4 persen. (Kesehatan, 2013).

Gambar 1.1 menyajikan kecenderungan status gizi dewasa menurut komposit TB dan IMT. Terlihat tidak terlalu banyak perubahan status gizi normal dari tahun 2007 ke tahun 2013 (<40%), selebihnya adalah variasi masalah pendek-gemuk, serta normal-gemuk. Terlihat kecenderungan meningkat untuk pendek gemuk, dan normal gemuk.



Gambar 1.1 Kecenderungan status gizi penduduk dewasa (diatas 18 tahun) berdasarkan komposit TB dan IMT Indonesia 2010 – 2013

Dalam survei yang diperoleh dapat disimpulkan bahwa setiap tahunnya grafik status gizi buruk orang Indonesia cenderung meningkat. Meskipun tidak terlalu signifikan namun permasalahan ini tentunya dapat menjadi serius jika tidak mendapat perhatian dan penanganan lebih lanjut.

Permasalahan dalam pemenuhan kebutuhan gizi ini pernah diangkat dengan menggunakan beberapa algoritma seperti pemrograman linear, *fuzzy Integer Programming* dan algoritma lainnya. Namun dalam kenyataannya masih belum menghasilkan hasil yang optimal. Hal inilah yang kemudian menjadi pertimbangan untuk diadakannya penelitian lebih lanjut terkait optimasi permasalahan pemenuhan gizi ini. Pemecahan persoalan pemrograman linier dengan metode grafis menemui kesulitan apabila dihadapkan pada persoalan yang mengandung lebih dari dua variabel. Untuk menanggulangi hal tersebut diperlukan suatu metode lain yang dalam hal ini yang disebut dengan metode Simpleks.

Metode Simpleks adalah algoritma yang digunakan untuk mengoptimalkan fungsi objektif dan memperhatikan semua persamaan kendala. Metode Simpleks pertama sekali diperkenalkan oleh George B. Dantzig dari USA (1950) melalui bukunya *Pemrograman linear and Extension*, menyebutkan bahwa ide dari pemrograman linear ini berasal dari ahli matematika Rusia bernama L. V. Kantorovich yang pada tahun 1939 menerbitkan sebuah karangan yang berjudul "*Mathematical Methods in the Organization and Planning of Production*".

Dengan memanfaatkan metode Simpleks ini, diharapkan permasalahan optimasi biaya gizi dan nutrisi harian dapat dipecahkan secara lebih aplikatif. Metode ini diharapkan dapat menghasilkan keluaran berupa menu harian yang optimal serta proporsional terhadap komposisi menu aslinya dan juga harga yang terjangkau.

1.2 Rumusan masalah

1. Bagaimana menerapkan metode Simpleks Big-M untuk optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa?

2. Bagaimana pengaruh parameter nilai minimum konsumsi dan jumlah banyak makanan dalam penerapan metode Simpleks Big-M yang digunakan untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa?
3. Bagaimana pengaruh parameter jenis dan jumlah makanan dalam penerapan metode Simpleks Big-M yang digunakan untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa?

1.3 Tujuan

1. Dapat menerapkan metode Simpleks Big-M untuk mengoptimasi biaya pemenuhan gizi dan nutrisi harian pada orang dewasa.
2. Dapat mengetahui pengaruh parameter nilai minimum konsumsi dalam penerapan metode Simpleks Big-M untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa.
3. Dapat mengetahui pengaruh parameter jenis dan jumlah makanan dalam penerapan metode Simpleks Big-M untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa.

1.4 Manfaat

1. Menyediakan solusi permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa yang lebih aplikatif.
2. Memberikan wawasan kepada pengguna aplikasi bahwa menu-menu makanan yang ada dapat dipadu-padankan untuk memenuhi gizi harian dengan harga terjangkau.

1.5 Batasan masalah

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

1. Sistem dirancang untuk diaplikasikan pada orang dewasa dengan usia minimal 15 tahun atau lebih.
2. Faktor yang mempengaruhi dibatasi jenis kelamin, usia, tinggi badan, berat badan dan tingkat aktivitas
3. Kebutuhan nutrisi yang digunakan dalam perhitungan adalah jumlah kalori dan *macronutrient*(protein , lemak dan karbohidrat).
4. Terdapat 174 macam pilihan makanan yang dapat dioptimasi.
5. Biaya pengeluaran optimum ditentukan dari komposisi jumlah makanan yang telah dipilih.
6. Pengujian sistem terdiri dari 3 macam pengujian yaitu : pengujian validasi, pengujian sensitivitas nilai minimum konsumsi dan pengujian sensitivitas jenis dan jumlah pilihan makanan.

1.6 Sistematika pembahasan

Penyusunan laporan tugas akhir ini dilakukan secara sistematis dan tersusun dalam tujuh bab dengan penjelasan sebagai berikut :

1. BAB I PENDAHULUAN

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

2. BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi uraian mengenai teori-teori yang dijadikan sebagai acuan dalam penulisan tugas akhir. Dasar teori yang dibahas meliputi diet problem, acuan kecukupan gizi dan metode Simpleks.

3. BAB III METODOLOGI

Bab ini menjelaskan mengenai metode dan langkah-langkah yang digunakan dalam penerapan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa dengan menggunakan metode Simpleks Big-M.

4. BAB IV PERANCANGAN

Bab ini berisi perancangan untuk membuat sistem. Adapun perancangan yang dibahas meliputi perancangan algoritma, perhitungan manual, perancangan database, perancangan antarmuka dan perancangan pengujian.

5. BAB V IMPLEMENTASI

Bab ini berisi implementasi sistem meliputi perangkat keras dan perangkat lunak yang digunakan, implementasi antar muka, struktur data serta pembahasan beberapa source code utama.

6. BAB VI PENGUJIAN DAN ANALISIS

Bab ini berisi pembahasan mengenai pengujian sistem serta analisa hasil pengujian.

7. BAB VII PENUTUP

Bab ini berisi kesimpulan dari seluruh rangkaian penelitian disertai dengan saran mengenai peluang pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian pustaka

Ada banyak metode yang dapat digunakan untuk memecahkan masalah optimasi gizi. Metode paling sederhana adalah dengan *Pemrograman linear*. Metode ini diawali dengan memodelkan masalah yang akan dioptimasi beserta batasan-batasannya ke dalam bentuk persamaan linier. Kemudian pemrograman linear akan berusaha mencari jumlah penyajian tiap bahan makanan sehingga didapatkan komposisi bahan yang optimal, yakni kandungan gizi maksimal dan biaya yang minimal.

Dalam penelitiannya "*Menu Planning with fuzzy 0-1 Integer Programming*" O.O. Kenan dkk menggunakan metode *fuzzy integer programming*. Model menu *planning* yang digunakan adalah *non-selective* yang artinya pemilihan makanan telah ditentukan oleh sistem melalui rule-rule yang ada. FIP tidak menentukan jumlah penyajian, hanya memilih makanan mana yang akan disajikan. Pada metode ini, hasil optimasi adalah *palatable* (enak dimakan) karena FIP hanya memilih apakah makanan disajikan atau tidak dengan jumlah penyajian tiap makanan yang telah ditentukan sebelumnya (Oruç, et al., 2012).

Beberapa referensi lainnya yang digunakan penulis untuk menyelesaikan permasalahan diet adalah dari segi pengoptimalan biaya. Dalam penelitian "*Desain dan Implementasi Sistem Informasi Pengelolaan Administrasi Biaya Proyek Konstruksi Bangunan dengan Metode Simpleks*" Robert Adi Santoso menggunakan metode Simpleks untuk menyelesaikan permasalahannya. Pemodelan permasalahan ke dalam bentuk linear dan penyelesaian masalah dalam bentuk metode Simpleks digunakan untuk mencari berapakah pekerja bangunan yang harus dipekerjakan tiap harinya sehingga didapatkan penghematan dari segi pengeluaran administrasi karena jumlah pekerja tidak berlebihan maupun terlalu sedikit. Dengan metode Simpleks ini, peneliti mampu mengoptimalkan penghematan biaya hingga sampai 77,27% (Santoso, 2013).

Dari beberapa referensi pendukung dari penelitian yang sebelumnya, maka penulis menyimpulkan bahwa dengan model permasalahan yang serupa, metode Simpleks dapat digunakan untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi harian bagi orang dewasa.

2.2 Diet problem

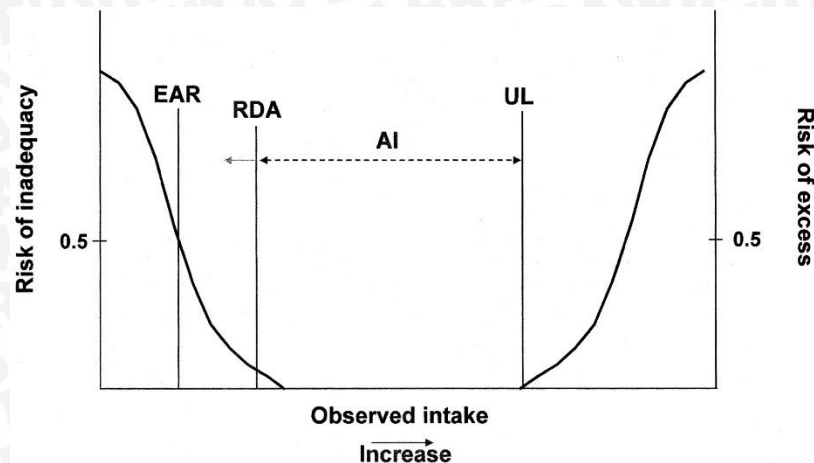
Permasalahan pemenuhan kebutuhan gizi atau dikenal pula dengan diet problem merupakan salah satu permasalahan klasik yang mulai dipelajari pada tahun 1940-an. Berawal dari pemenuhan diet sehat bagi tentara Amerika, diet problem kemudian berkembang hingga saat ini. Menurut Matthew Ford, tujuan utama pemecahan diet problem adalah untuk menemukan kombinasi makanan termurah yang dapat memenuhi kebutuhan nutrisi harian minimal seseorang.

2.3 DRI dan AKG

Sebelum dapat menentukan apakah suatu *diet plan* dapat memenuhi kebutuhan nutrisi harian seseorang, perlu ditentukan terlebih dahulu jumlah tiap nutrient 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 DRI(*Dietary Reference Intakes*). Di Indonesia sendiri diterapkan standar yang disebut dengan AKG(Angka Kecukupan Gizi).

DRI adalah standar yang dikembangkan secara kolaboratif oleh *Food and Nutrition Board* dari *Institute of Medicine* Amerika dan Kanada. DRI adalah seperangkat nilai-nilai berbasis nutrisi yang dapat digunakan untuk mengevaluasi diet. Yang termasuk di antaranya adalah :

- a. *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 dibawah 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.
- b. *Adequate Intake*(AI) adalah angka pendekatan yang diasumsikan akan dapat memenuhi kebutuhan nutrisi seseorang untuk tetap sehat. Angka ini ditetapkan jika tidak ada informasi yang cukup untuk menentukan RDA.
- c. *Tolerable Upper Intake Level* atau *Upper Level*(UL) ditetapkan untuk beberapa vitamin dan mineral, UL adalah jumlah tertinggi asupan sebuah nutrient agar tidak menimbulkan efek merugikan bagi kesehatan dalam jangka panjang. Semakin jauh asupan melebihi nilai UL, resiko efek merugikan semakin naik.



Gambar 2.1 hubungan RDA, AI dan UL

- d. *Estimated Energy Requirement* (EER) adalah angka yang digunakan untuk menentukan kebutuhan kalori. Angka RDA ditetapkan sedikit lebih tinggi dari kebutuhan nutrisi rata-rata. Hal ini tidak masalah untuk nutrisi karena kelebihan sedikit nutrisi 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.
- e. *Acceptable Macronutrient Distribution Range* (AMDR) adalah rangkaian asupan untuk sumber energi khusus (protein, lemak, karbohidrat), dinyatakan sebagai persentase dari total energi, yang dihubungkan dengan penurunan risiko penyakit kronis sekaligus memberikan asupan yang memadai kebutuhan nutrisi penting. Berdasarkan Data DRI *Macronutrient* Untuk remaja dan Dewasa memiliki nilai AMDR 45-65% untuk karbohidrat, 20 -35% untuk lemak dan 10-35% untuk protein. Sedangkan untuk anak-anak memiliki nilai AMDR yang bervariasi berdasarkan usianya.

Di Indonesia, acuan kecukupan gizi yang digunakan adalah Angka kecukupan Gizi (AKG). Angka ini ekuivalen dengan RDA. Nilai-nilai pada AKG menunjukkan rata-rata jumlah nutrisi yang dianjurkan untuk dikonsumsi oleh individu dalam suatu kelompok usia dan jenis kelamin per harinya. Nilai pada AKG hanya berlaku bagi orang sehat dan kondisi khusus yaitu ibu hamil dan ibu menyusui semester pertama dan kedua. Di Indonesia, AKG disusun dan terus 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 tingkat regional maupun Internasional.

2.4 Harris benedict equation

Harris Benedict Equation adalah metode yang digunakan untuk mengestimasi Individual Basal Metabolic Rate(BMR) dan kebutuhan sehari-hari per kilokalori. Nilai BMR ini kemudian dipengaruhi tidak hanya usia,tinggi badan,berat badan dan jenis kelamin. Namun nilai BMR juga dipengaruhi oleh level aktivitas individu. Rumus harris benedict ini ditemukan oleh James Arthur Harris dan Franciss Gano Benedict pada tahun 1919. Persamaan BMR ini kemudian di revisi pada tahun 1984 yang mana revisi ini menghasilkan akurasi yang lebih besar daripada persamaan sebelumnya. Adapun Persamaan BMR yang telah direvisi ditunjukkan dalam Persamaan 2.1 dan 2.2 (Wahyuningsih, 2013).

Perhitungan BMR untuk Pria

$$\text{BMR} = 88.362 + (13.397 \times \text{berat (kg)}) + (4.799 \times \text{tinggi (cm)}) - (5.677 \times \text{usia (tahun)}) \quad (2.1)$$

Perhitungan BMR untuk Wanita

$$\text{BMR} = 447.593 + (9.247 \times \text{berat (kg)}) + (3.098 \times \text{tinggi (cm)}) - (4.330 \times \text{usia (tahun)}) \quad (2.2)$$

Setelah mendapatkan BMR tiap harinya maka langkah selanjutnya adalah menentukan nilai Recommended Intake berdasarkan nilai Recommended Intake ini dalam Rumus juga disebut Total Energy Expenditure (total keluaran energi)/TEE. TEE ini dihitung berdasarkan tingkat aktivitas dari individu. Tingkat aktivitas dibagi menjadi 5 dengan kategori sebagai berikut.

Tabel 2.1 Tingkat aktivitas TEE

Level Aktivitas	Kalori Harian yang Dibutuhkan
Sedentary : Untuk aktivitas ringan atau tidak beraktivitas.	BMR x 1.2
Mild activity level : Latihan intensive selama kurang lebih 20 menit. 1 sampai 3 kali per minggu. Contoh aktivitas ini adalah bersepeda, jogging, basket, berenang, dan lain-lain. Jika tidak melakukan latihan rutin, tetapi memiliki kehidupan yang sibuk yang membutuhkan anda sering berjalan untuk periode waktu yang lama, Anda harus memilih jenis level ini.	BMR x 1.3 - 1.375
Moderate activity level : Latihan intensive selama kurang lebih 30 sampai 60 menit. 3 sampai 4 kali per minggu. Contoh aktivitas	BMR x 1.5 - 1.55

level ini sama seperti aktivitas pada Mild Activity level.	
Heavy or (Labor-intensive) activity level : Latihan intensive selama 60 menit atau lebih. 5 sampai 7 kali per minggu (untuk contoh aktivitas seperti level Mild Activity). Pekerjaan berat secara intensive juga memenuhi syarat untuk level ini. Pekerjaan berat ini meliputi pekerjaan kontruksi (tukang batu, tukang kayu, tukang pada umumnya, dan lain lain.). Juga petani atau pekerjaan yang serupa.	BMR x 1.7
Extreme level : Aktivitas yang sangat aktif dan atau aktivitas yang menuntut: contoh yang termasuk aktivitas ini: atlet dengan jadwal latihan yang tak padat dengan banyak sesi latihan sepanjang hari atau beberapa pekerjaan yang menuntut, seperti penambang batubara atau bekerja pada waktu yang lama. Umumnya level aktivitas ini sulit untuk di capai.	BMR x 1.9

2.5 Pemrograman linear

Dalam istilah matematika, pemrograman linear 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. Pemrograman linear adalah salah satu teknik *operations research* yang menjadi dasar berkembangnya teknik *operations research* yang lain seperti : *Goal Programming*, *Binary Integer Programming* atau *Zero-one Programming*. pemrograman linear juga biasa dipadukan dengan teknik *operations research* yang lain yang digunakan untuk menyelesaikan permasalahan-permasalahan tertentu. pemrograman linear memiliki bentuk umum yang ditunjukkan pada persamaan 2.3 dan 2.4 (Levin, 2012) :

Maksimumkan atau minimumkan :

$$C_1X_1 + C_2X_2 + \dots + C_nX_n \quad (2.3)$$

Dengan batasan :

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq, = \text{atau} \geq b_1 \\
 a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq, = \text{atau} \geq b_2 \\
 a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq, = \text{atau} \geq b_m \\
 x_1, x_2, \dots, x_n &\geq 0
 \end{aligned} \quad (2.4)$$

Model adalah sebuah tiruan terhadap realitas. Langkah untuk membuat peralihan dari realita ke model kuantitatif, dinamakan perumusan model. Untuk memodelkan permasalahan ke dalam bentuk linear programming perlu diperhatikan 3 unsur berikut, yaitu :

1. Variabel Keputusan
2. Fungsi Tujuan
3. Fungsi Kendala

Variabel Keputusan adalah variabel persoalan yang akan mempengaruhi nilai tujuan yang hendak dicapai. Dalam proses pemodelan, penemuan variabel keputusan tersebut harus dilakukan terlebih dahulu sebelum merumuskan fungsi tujuan dan kendala-kendalanya.

Fungsi Tujuan adalah sebuah target yang ingin dicapai dan diwujudkan dalam bentuk fungsi persamaan linear. Fungsi ini kemudian dimaksimumkan atau diminimumkan terhadap kendala-kendala yang ada. Dalam beberapa permasalahan fungsi tujuan biasanya dimaksimumkan apabila ingin mendapatkan laba/keuntungan sedangkan fungsi tujuan diminimumkan apabila ingin meminimalisir pengeluaran/biaya produksi.

Fungsi Kendala merupakan suatu pembatas terhadap kumpulan keputusan yang mungkin dibuat untuk mempengaruhi fungsi tujuan dan harus diwujudkan dalam bentuk fungsi persamaan linear. Dalam hal ini, sesuai dengan dalil-dalil matematika, ada tiga macam kendala yaitu:

1. Kendala berupa pembatas
2. Kendala berupa syarat
3. Kendala berupa keharusan

Kendala yang berupa pembatas mengendalikan nilai ruas kiri agar tidak lebih besar dari nilai ruas kanannya. Kendala berupa pembatas dituangkan ke dalam fungsi matematika yang berupa pertidaksamaan dengan tanda " $\leq$ ".

Kendala yang berupa syarat mengendalikan nilai ruas kiri agar tidak lebih kecil daripada nilai pada ruas kanannya. Kendala berupa syarat dituangkan ke dalam fungsi matematika yang berupa pertidaksamaan dengan tanda " $\geq$ ".

Kendala yang berupa keharusan mengendalikan nilai ruas kiri agar nilainya sama dengan nilai pada ruas kanannya. Kendala berupa keharusan dituangkan ke dalam fungsi matematika yang berupa persamaan dengan tanda " $=$ ".

2.6 Metode Simpleks

Metode Simpleks adalah sebuah prosedur matematis berulang untuk menemukan penyelesaian optimal soal pemrograman linear dengan cara menguji titik-titik sudutnya. Seperti pengertiannya sendiri bahwa metode Simpleks digunakan untuk menyelesaikan bentuk dari soal pemrograman linear yang artinya bentuk tersebut bisa berupa persamaan ataupun pertidaksamaan. Setiap soal linear programming akan memiliki tanda kendala sekurang-kurangnya satu atau lebih dari tanda $\geq$, $=$ atau $\leq$. Tanda kendala ketidaksamaan ($\leq$ dan $\geq$) inilah yang

harus diubah dahulu ke bentuk kesamaan(=). Perubahan tanda kendala dapat dilakukan dengan menggunakan tambahan variabel. Variabel tambahan ini kemudian memiliki peranannya sendiri dalam penggunaannya. Variabel-variabel tersebut adalah :

a. Variabel Slack

Variabel *Slack* adalah variabel tambahan yang berfungsi untuk menampung sisa kapasitas pada kendala yang berupa pembatas ($\leq$). Sebagai contoh adalah persamaan $X_1 + X_2 \leq 5$. Maka untuk mengubah pertidaksamaan diatas menjadi sebuah persamaan, harus ditambahkan variabel baru dimana disini disebut *slack* variabel. Kita misalkan variabel baru tersebut dengan “ S ”, sehingga bentuk persamaannya menjadi $X_1 + X_2 + S = 5$. Tabel 2.2 menunjukkan nilai variabel slack dalam beberapa titik.

Tabel 2.2 Variabel Slack

Koordinat			Koordinat		
X_1	X_2	S	X_1	X_2	S
1	1	3	5	0	0
1	2	2	0	5	0
2	2	1	2	3	0
1	3	1	1	4	0

Jika kita melihat Tabel 2.2, maka variabel S tersebut berfungsi untuk membuat nilai ruas kiri sama dengan nilai ruas kanannya. Semakin jauh (X_1, X_2) dari garis kendala semakin besar nilai nilai S, dan sebaliknya semakin dekat (X_1, X_2) ke garis kendala semakin kecil nilai S dan akan mencapai nol pada saat tepat berada di garis kendala. Variabel S ini juga berfungsi untuk menampung sisa kapasitas yang tidak digunakan. Sebagai contoh sisa kapasitas adalah 2 untuk $X_1 = 1$ dan $X_2 = 2$. Sedangkan, total kapasitas yang seharusnya digunakan adalah $X_1 = 2$ dan $X_2 = 3$ atau $X_1 = 3$ dan $X_2 = 2$.

b. Variabel Surplus

Variabel *Surplus* adalah variabel tambahan yang berfungsi untuk menampung kelebihan nilai ruas kiri pada kendala yang berupa syarat ($\geq$). Sebagai contoh adalah persamaan $X_1 + X_2 \geq 5$. Perbedaan antara persamaan (didas dengan persamaan pada *slack* variabel) adalah pada persamaan(didas) nilai ruas kiri tidak boleh lebih kecil daripada nilai ruas kanan. Agar nilai ruas kirinya sama dengan nilai ruas kanannya, kita perlu menghadirkan sebuah variabel baru dimana kita misalkan dengan notasi “S” yang berkoefisien “-1”. Variabel ini berfungsi menampung kelebihan nilai ruas kiri. Sehingga bentuk pertidaksamaan tersebut menjadi bentuk

persamaan. Berikut model persamaannya menjadi $X_1 + X_2 - S = 5$. Tabel 2.3 menunjukkan nilai variabel *surplus* dalam beberapa titik.

Tabel 2.3 Variabel Surplus

Koordinat			Koordinat		
X_1	X_2	S	X_1	X_2	S
5	3	3	5	0	0
4	3	3	0	5	0
4	3	2	2	3	0
3	3	1	1	4	0

Dari Tabel tersebut kita dapat menyimpulkan bahwa variabel S berfungsi untuk menampung kelebihan nilai ruas kiri agar sama dengan nilai ruas kanan atau kapasitasnya. Bila (X_1, X_2) tepat berada pada garis kendala, maka S akan bernilai nol dan *surplus* itu tidak ada. Sebagai contoh S adalah 0 untuk $X_1 = 5$ dan $X_2 = 0$.

c. Variabel *Artificial*

Variabel *Artificial* adalah variabel tambahan yang ditambahkan pada kendala yang berupa syarat ($\geq$) atau kendala yang berupa keharusan ($=$). Variabel ini diperlukan sebagai peubah karena fungsinya yang seperti *slack* variabel. Dalam setiap iterasi penyelesaian dengan metode Simpleks diperlukan matriks basik. Untuk fungsi kendala dengan tanda ketidaksamaan ($\geq$) dapat diubah menjadi fungsi kendala dengan tanda kesamaan ($=$) dengan menambahkan variabel *surplus*. Variabel *surplus* ini memiliki koefisien “-1” sehingga akan menimbulkan kesulitan dalam memecahkan soal dengan metode Simpleks. Sebagai contoh adalah fungsi kendala $2X_1 \geq 80$. Penambahan variabel surplus ke dalam fungsi kendala ini menyebabkan fungsi kendala berubah menjadi $2X_1 - S_1 = 80$.

S_1 adalah variabel surplus. Jika $X_2 = 0$, maka $S_1 = -80$. Ini berarti bahwa pada pemecahan pertama, fungsi kendala ini sudah tidak fisibel. Kondisi semacam ini tidak diperbolehkan dalam memecahkan soal pemrograman linear dengan metode Simpleks. Untuk mengatasi masalah ini, digunakan variabel buatan atau *artificial*(A) yang harus ditambahkan di fungsi kendala dengan tanda $\geq$. Sehingga persamaannya menjadi $2X_1 - S_1 + A_1 = 80$. Dalam fungsi tujuan, variabel *surplus* ini ditambahkan dengan koefisien nol(0). Demikian pula dengan variabel *artificial* harus ditambahkan ke fungsi tujuan dengan koefisien satu(1).

2.6.1 Variabel basis dan nonbasis

Variabel basis dan nonbasis adalah dua terminologi penting yang akan selalu digunakan di dalam metode Simpleks. Variabel basis adalah variabel yang bernilai positif di sembarang iterasi dan variabel nonbasis adalah variabel yang bernilai nol pada sembarang iterasi. Pada solusi awal, Variabel basis merupakan variabel slack (jika fungsi kendala menggunakan pertidaksamaan $\leq$) atau variabel buatan (jika fungsi kendala menggunakan pertidaksamaan $\geq$ atau $=$). Sedangkan variabel nonbasis merupakan variabel keputusan pada kendala-kendala yang ada. Sebagai contoh adalah :

- a. $2X_1 + 2X_2 + S_1 = 20$
- b. $2X_1 + 5X_2 + S_2 = 30$
- c. $2X_1 + 7X_2 - S_3 = 40$

Dari persamaan diatas, X_1 dan X_2 dinamakan variabel nonbasis karena dapat bernilai nol, sedangkan S_1 , S_2 , dan S_3 dinamakan variabel basis.

2.6.2 Metode Big-M

Dalam beberapa permasalahan yang diwujudkan dalam bentuk fungsi kendala. Sering ditemukan pertidaksamaan yang tidak hanya menggunakan ($\leq$), akan tetapi sering dijumpai beberapa fungsi kendala dengan pertidaksamaan ($\geq$) atau bentuk persamaan ($=$). Fungsi kendala dengan tanda ($\geq$) hanya memiliki surplus variabel dan tidak memiliki slack variabel. Begitupula dengan Fungsi kendala ($=$) yang bahkan tidak memiliki surplus ataupun *slack* variabel. *Surplus* variabel tidak dapat digunakan sebagai variabel basis awal, dengan demikian harus ditambahkan lagi satu variabel baru yang dapat berfungsi sebagai variabel basis awal. Variabel yang bisa menjadi Variabel basis awal hanya slack variabel dan *artificial* variabel. Dengan penambahan *artificial* variabel ini maka penyelesaian selanjutnya adalah dengan metode Big-M. Pada metode Big-M ini *artificial* variabel diberikan suatu penalti dengan suatu bilangan yang besar sekali pada fungsi tujuan. Metode Simpleks kemudian mencoba untuk memperbaiki fungsi tujuan dengan cara membuat *artificial* variabel tidak ekonomis lagi untuk dipertahankan sebagai variabel basis dengan nilai yang positif di setiap iterasinya. Fungsi Tujuan dalam metode Simpleks Big-M sendiri dibagi menjadi dua :

1. Fungsi Minimisasi
2. Fungsi Maksimisasi

Perbedaan keduanya adalah pada fungsi tujuan minimisasi akan ada penambahan sejumlah M koefisien pada *artificial* variabelnya. Sedangkan, pada fungsi tujuan maksimisasi akan ada penambahan sejumlah -M koefisien pada *artificial* variabelnya.

2.6.3 Penerapan metode Simpleks Big-M

Dalam membuat perhitungan dengan Algoritma ini. Maka terdapat beberapa tahap penyelesaian. Berikut Tahap-tahap penyelesaian metode Simpleks metode Big-M.

Tahap 1 : Formulasikan masalah

Memformulasikan masalah ke dalam soal pemrograman linear adalah langkah awal. Dengan menentukan fungsi tujuan dan kendala dari permasalahan maka akan mudah untuk selanjutnya menentukan penambahan variabel basis maupun nonbasisnya.

Tahap 2 : Pembuatan tabel awal

Untuk memecahkan soal pemrograman linear yang telah ditentukan pada tahap sebelumnya ada dua hal yang perlu diperhatikan. Pertama, perubahan fungsi tujuan dan fungsi kendala dengan menambahkan variabel *slack* atau *surplus* dan variabel *artificial*. Kedua, menyusun tabel penyelesaian. Untuk model maksimisasi, tambahan variabel buatan diberikan koefisien dengan laba RpM yang bernilai sangat besar. Sedangkan untuk model minimisasi, tambahan variabel buatannya diberikan koefisien biaya RpM yang juga bernilai sangat besar.

Tahap 3 : Penentuan kolom kunci

Menentukan baris kunci dilakukan dengan melihat baris $C_j - Z_j$. Untuk model minimisasi, variabel yang dipilih dilihat dari negatif $C_j - Z_j$ yang terbesar. Adapun untuk model maksimisasi, variabel yang dipilih dari nilai positif $C_j - Z_j$ yang terbesar.

Tahap 4 : Penentuan baris kunci

Menentukan baris kunci dilakukan dengan cara membagi nilai kuantitas dengan koefisien yang sebaris dengan baris pada kolom kunci. Variabel baris yang akan diganti ditentukan oleh hasil pembagian positif terkecil.

Tahap 5 : Perhitungan koefisien baru untuk variabel yang masuk

Menentukan koefisien baru untuk variabel yang masuk dengan menggunakan koefisien elemen perpotongan. Pada tahap ini elemen baris yang terpilih/baris kunci pada tahap 4 dibagi dengan koefisien elemen perpotongan terpilih. Sehingga menghasilkan koefisien baris baru.

Tahap 6 : Revisi koefisien baris lainnya

Langkah selanjutnya adalah merevisi koefisien baris-baris kendala yang lain dengan menggunakan metode pivot atau metode substitusi. Pada tahap ini, dengan menggunakan koefisien baris baru terpilih pada tahap 5, kemudian digunakan untuk mencari koefisien baris baru pada kendala-kendala yang lain.

Tahap 7 : Menyusun tabel jawaban

Susunlah tabel jawaban berdasarkan koefisien-koefisien yang dihitung pada tahap ke-5 dan ke-6.

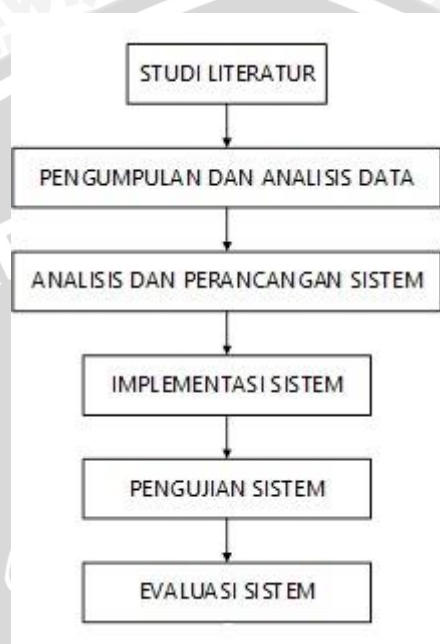
Tahap 8 : Periksa tabel jawaban

Memeriksa tabel jawaban diperlukan untuk mengetahui apakah jawaban telah optimal. Jika peningkatan nilai Z masih dapat dilakukan, ulangi lagi prosedur di atas dimulai dari tahap ketiga (Muslich, 2010).

BAB 3 METODOLOGI

3.1 Tahapan penelitian

Tahapan penelitian ini meliputi studi literatur, pengumpulan dan analisis data, analisis dan perancangan sistem, implementasi sistem, pengujian sistem, dan evaluasi hasil yang ditunjukkan pada Gambar 3.1 berikut ini.



Gambar 3.1 Tahap penelitian

Berdasarkan Gambar 3.1 penjabaran langkah-langkah yang dilakukan dalam penelitian ini adalah :

1. **Studi literatur**

Mencari dan mempelajari literatur yang berhubungan dengan penerapan metode *Simpleks Big-M* dalam menyelesaikan masalah optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa.

2. **Pengumpulan dan analisis data**

Mengumpulkan data makanan beserta kandungan gizinya dan Harga makanan. Data Kandungan gizi bersumber dari data Departemen Kesehatan tahun 2010. Sedangkan untuk harga didapatkan dari penelitian (Suci, 2015).

3. **Analisis dan perancangan sistem**

Menganalisis dan merancang sistem penerapan metode *Simpleks Big-M* dalam menyelesaikan masalah optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa.

4. **Implementasi sistem**

Mengimplementasikan hasil analisa dan perancangan sistem dengan membangun suatu perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman yaitu dengan menggunakan bahasa pemrograman berbasis web dan tools pendukung lainnya. Implementasi Optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa meliputi penerapan metode Simpleks Big-M dalam program yang dibuat dengan berbasis web, pembuatan antar muka program dan pengujian sistem.

5. Pengujian sistem

Pada tahap ini dilakukan proses pengujian sistem dengan menggunakan data yang telah tersedia. Data akan dimasukan ke sistem dan diproses oleh sistem menggunakan perhitungan metode Simpleks Big-M. Pengujian perangkat lunak pada penelitian ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhannya. Dalam penelitian ini akan menguji kesesuaian output dengan fungsi kendalanya dan keoptimalan hasil dengan fungsi tujuannya(minimum).

6. Evaluasi sistem

Langkah terakhir dalam penelitian ini adalah menganalisis hasil uji coba metode yang telah diterapkan dan mengevaluasinya. Tujuannya adalah mengetahui nilai-nilai parameter metode yang terbaik dalam optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa. Tahapan evaluasi ini juga dilakukan pada perangkat lunak yang digunakan untuk mengukur performa dari sistem yang telah dibuat dan guna mengetahui kekurangan-kekurangan sementara dari sistem tersebut. Dari sini dapat diketahui apakah sistem yang dibuat telah sesuai dengan tujuan awal dari dilakukannya penelitian.

3.2 Kebutuhan sistem

Kebutuhan sistem merupakan tahap menganalisis hal-hal yang dibutuhkan sistem sebelum mengimplementasikannya ke dalam kode program.

3.2.1 Deskripsi umum sistem

Sistem yang dibuat merupakan perangkat lunak yang mengimplementasikan Simpleks Big-M dalam menyelesaikan masalah optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa. Sistem ini akan merekomendasikan gram komposisi makanan yang optimal untuk memenuhi kebutuhan nutrisi dari individu serta meminimalisir pengeluaran biaya.

Sistem ini memerlukan data berupa daftar bahan makanan di Indonesia beserta kandungan nutrisi dalam setiap bahan makanan yang digunakan. Sistem ini memerlukan beberapa masukan dari pengguna yaitu jenis kelamin, usia , berat badan,tinggi badan,level aktivitas dan bahan makanan yang dipilih. Sistem diharapkan dapat memberikan keluaran berupa rekomendasi komposisi kuantitas bahan makanan yang dipilih dengan hasil paling optimal.

3.2.2 Data yang digunakan

Data yang digunakan dalam penelitian adalah sebagai berikut:

1. Daftar bahan makanan beserta keterangan nutrisi meliputi energi, protein, karbohidrat dan lemak didapatkan dari Departemen kesehatan.
2. Daftar harga bahan makanan didapatkan berdasarkan harga survey pasar kota Kediri pada bulan Januari 2015 yang dilakukan untuk penelitian Widya.W.S,2015.

3.3 Formulasi permasalahan

Optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa dengan menggunakan metode Simpleks Big-M terdiri dari beberapa proses sebagai berikut:

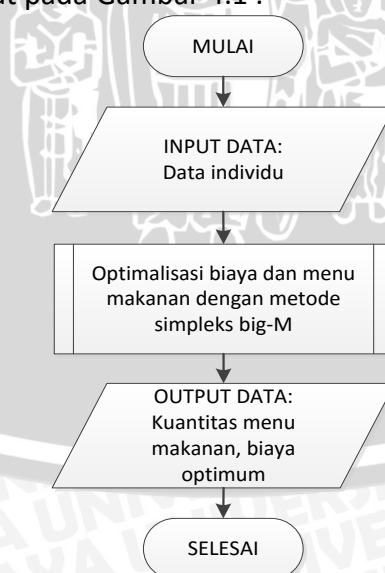
1. Melakukan proses perhitungan nutrisi (karbohidrat , lemak dan protein) dengan memasukkan data berupa :
 - Usia
 - Berat badan
 - Tinggi badan
 - Jenis kelamin
 - Jenis aktivitas
 - Makanan yang tersedia
2. Memformulasikan masalah ke dalam bentuk *pemrograman linear*.
3. Membuat tabel awal yang berisikan fungsi tujuan dan kendala-kendala dari bentuk *pemrograman linear*.
4. Memilih variabel pada kolom dengan dengan nilai $C_j - Z_j$ negatif terbesar sebagai variabel terpilih.
5. Menentukan variabel pada baris kendala yang akan diganti dengan variabel terpilih.
6. Menghitung koefisien baru untuk variabel terpilih.
7. Merevisi koefisien baris kendala lainnya.
8. Menyusun tabel Jawaban
9. Melakukan proses iterasi , iterasi selesai jika sudah tidak ada $C_j - Z_j$ yang bernilai negatif. Jika masih ada, maka dilakukan perhitungan dimulai dari langkah ke empat.

BAB 4 PERANCANGAN

4.1 Perancangan algoritma

Pada bagian ini awalnya dibahas proses perancangan sistem untuk memperoleh solusi keluaran yang baik. Langkah awal proses ini adalah melakukan perhitungan jumlah kalori dan menentukan kebutuhan *macronurient* (protein, lemak dan karbohidrat) dari user. Langkah selanjutnya adalah memodelkan fungsi tujuan dan fungsi kendala ke dalam bentuk *pemrograman linear*. Setelah memodelkan ke dalam bentuk linear programming maka selanjutnya adalah mengubah fungsi kendala dan fungsi tujuan tersebut ke dalam bentuk model standar. Pada model standar, fungsi-fungsi tersebut diberi variabel *surplus* dan variabel *artificial* atau variabel *slack*.

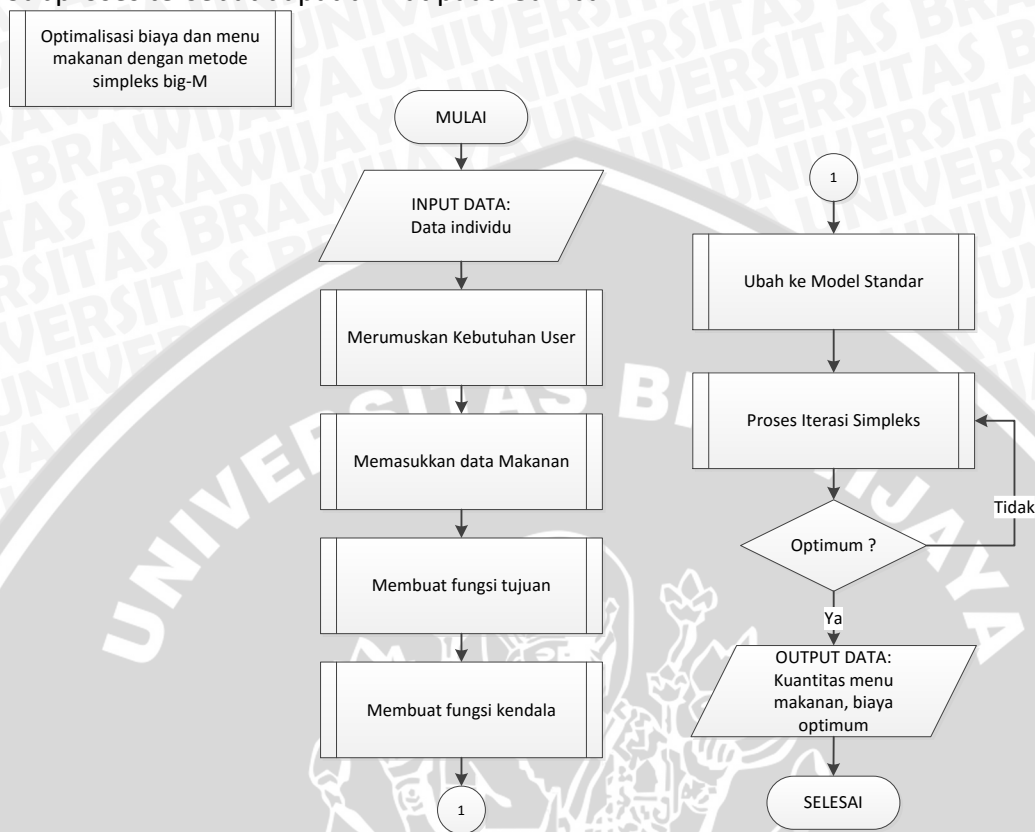
Setelah memodelkan ke bentuk standar berikutnya adalah mengubahnya ke dalam bentuk matriks metode Simpleks Big-M. Dari matriks ini kemudian dibuat perhitungan untuk mendapatkan nilai Z_j yang didapatkan dari $\sum$ dari kolom C_j dikalikan dengan masing-masing baris dari masing-masing kolom. Setelah itu dicari nilai kolom yang memiliki nilai $C_j - Z_j$ negatif terbesar untuk menjadi variabel terpilih. Kemudian menentukan variabel pada baris kendala yang akan diganti dengan variabel terpilih. Setelah itu menghitung koefisien baru untuk variabel terpilih untuk menggantikan variabel pada baris kendala yang terpilih dengan menggunakan nilai perpotongan antara baris dan kolom terpilih tersebut. Nilai koefisien baru ini kemudian digunakan untuk merevisi baris kendala lainnya. Selanjutnya dibuat kembali langkah tabel matriks dan sekaligus melakukan proses iterasi hingga tidak ada lagi nilai $C_j - Z_j$ yang negatif. Proses penyelesaian optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa menggunakan metode Simpleks Big-M dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram alir sistem

Flowchart pada Gambar 4.1 menjelaskan tentang optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa menggunakan metode Simpleks Big-M di mana

terdapat subproses di dalamnya yaitu proses optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa dengan menggunakan metode Simpleks Big-M. Subproses tersebut dapat dilihat pada Gambar 4.2.

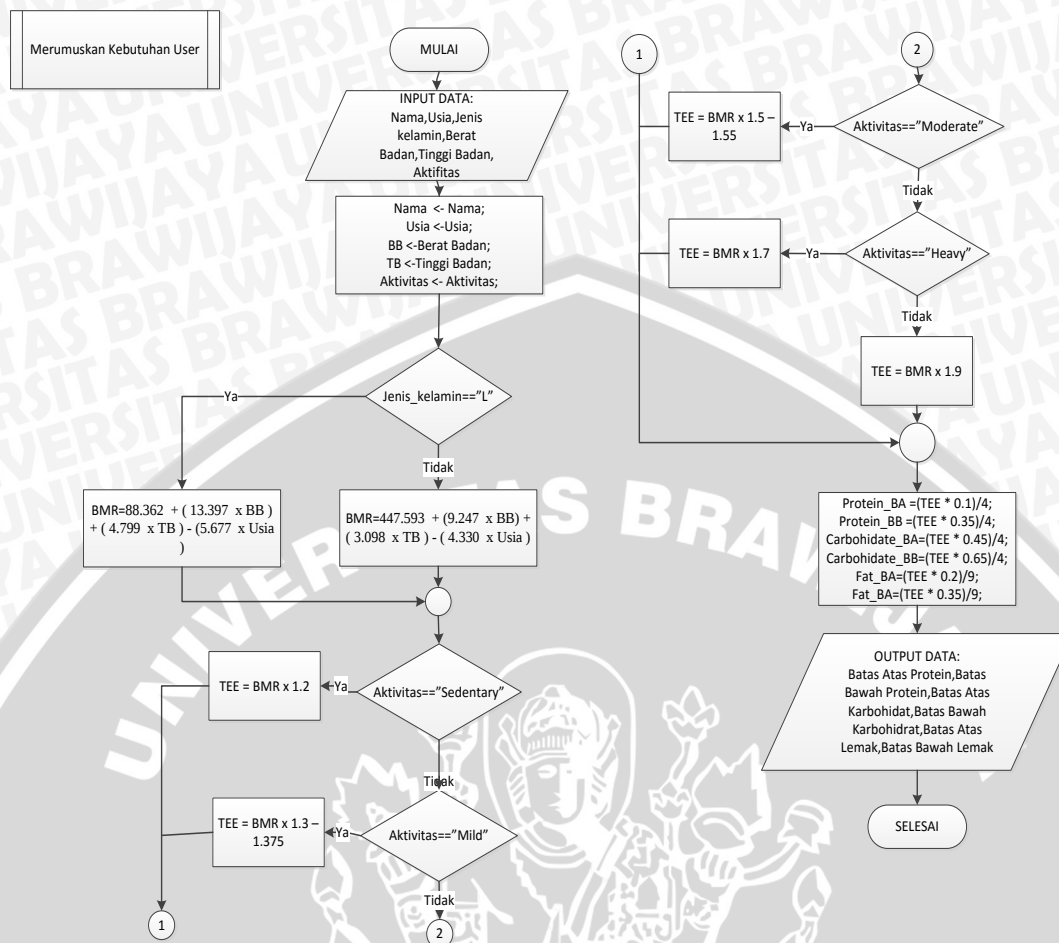


Gambar 4.2 Flowchart proses optimalisasi biaya dan menu makanan dengan metode Simpleks Big-M.

Seperti yang terlihat pada Gambar 4.2 terdapat tujuh subproses yang berisikan langkah-langkah penyelesaian antara lain : merumuskan kebutuhan nutrisi user, memasukkan data makanan yang dipilih oleh user, membuat/memodelkan fungsi tujuan, membuat/memodelkan fungsi kendala, mengubah fungsi tujuan dan fungsi kendala ke dalam bentuk model standar dan melakukan proses iterasi perhitungan Simpleks hingga ditemukan hasil yang paling optimum. Dari langkah –langkah tersebut maka output yang dikeluarkan adalah optimalnya kuantitas dari makanan yang harus dikonsumsi beserta biaya yang harus dikeluarkan oleh individu (biaya yang paling minimum).

4.1.1 Merumuskan kebutuhan user

Pada proses ini sistem akan menghitung kebutuhan macronutrient yang seharusnya dibutuhkan oleh individu. Proses perhitungan dengan rumus Harris Benedict Equation digunakan untuk menghitung individual *Basal Metabolic Rate*(BMR). Proses pengerjaannya dapat dilihat pada Gambar 4.3 berikut :



Gambar 4.3 Flowchart merumuskan kebutuhan user.

Berdasarkan Gambar 4.3 dapat dijelaskan langkah-langkah dalam merumuskan kebutuhan user adalah sebagai berikut :

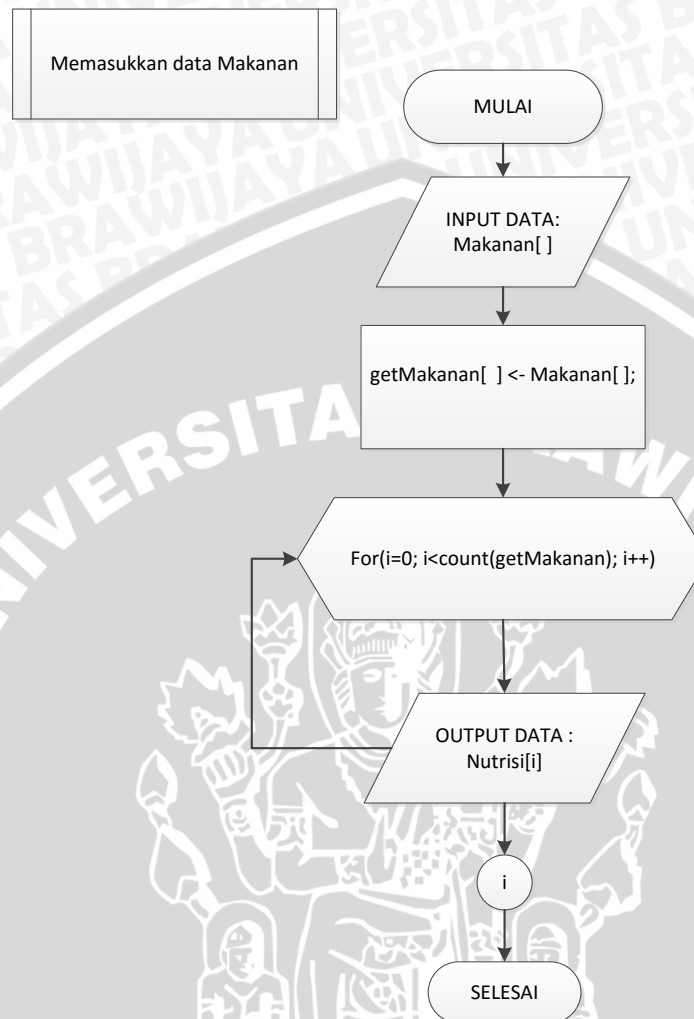
1. Sistem menerima masukan berupa data individu dan tingkat aktivitas sehari-hari
2. Sistem akan melakukan proses menghitung jumlah kalori yang dibutuhkan setiap harinya.
3. Sistem melakukan perhitungan *macronutrient* yang dibutuhkan berdasarkan kebutuhan kalori dari individu.

Hasil dari proses ini berupa batas atas dan bawah dari kebutuhan *macronutrient* yang dibutuhkan individu.

4.1.2 Memasukkan data makanan

Pada proses ini individu melakukan penginputan makanan-makanan apa saja yang akan dikonsumsi selama satu hari. Proses penginputan ini diambil dari

database makanan-makanan indonesia. Langkah-langkah proses memasukkan data makanan dapat dilihat pada Gambar 4.4 berikut :



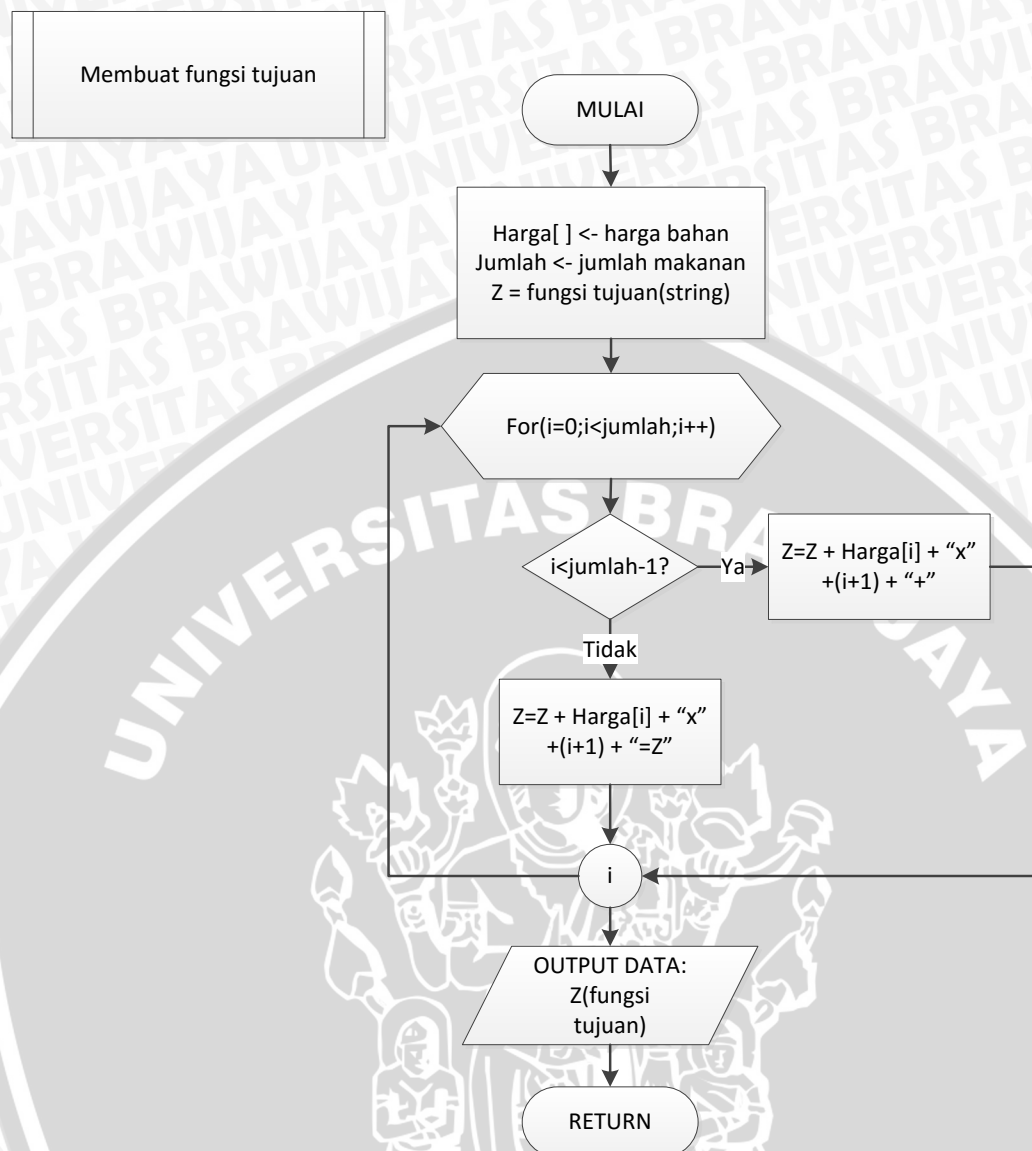
Gambar 4.4 Flowchart memasukkan data makanan.

Berdasarkan Gambar 4.4 dapat dijelaskan langkah-langkah dalam memasukkan data makanan adalah sebagai berikut :

1. Sistem akan menerima inputan berupa makanan yang ingin dikonsumsi oleh individu.
2. Sistem akan menampilkan makanan yang dipilih dalam bentuk tabel matriks yang berisi nama makanan, nutrisi dan harga.

4.1.3 Membuat fungsi tujuan

Pada proses ini berdasarkan inputan yang dilakukan pada proses sebelumnya, sistem mengubahnya menjadi bentuk pemrograman linear. Salah satunya adalah mengubah input menjadi fungsi tujuan. Langkah-langkah proses membuat fungsi tujuan dapat dilihat pada Gambar 4.5 berikut :



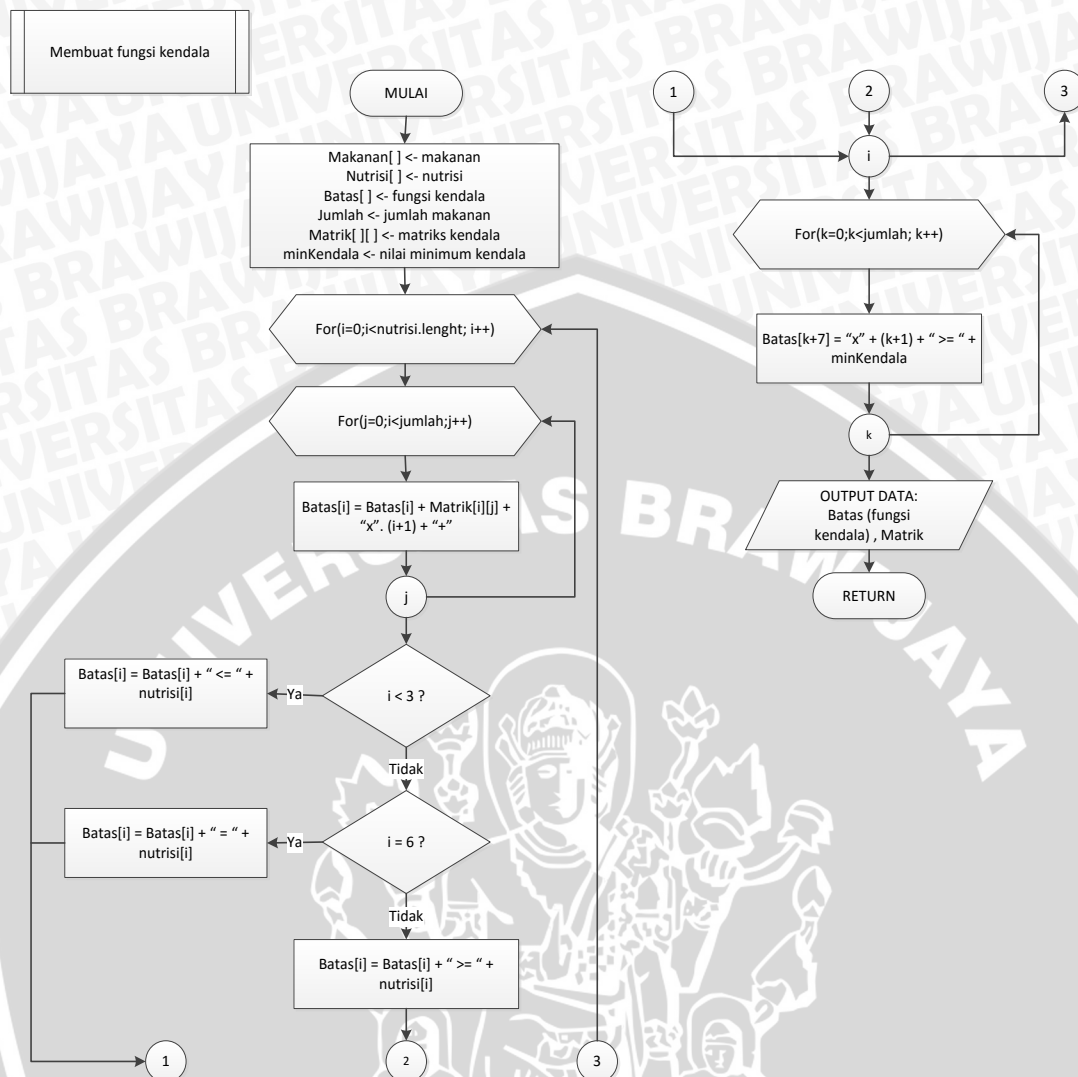
Gambar 4.5 Flowchart membuat fungsi tujuan.

Berdasarkan Gambar 4.5 dapat dijelaskan langkah-langkah dalam membuat fungsi tujuan adalah sebagai berikut :

1. Sistem akan mengubah dan memberi konstanta pada setiap makanan yang dipilih pada proses sebelumnya.
2. Sistem akan menampilkan bentuk dari Z(fungsi tujuan) .

4.1.4 Membuat fungsi kendala

Pada proses ini berdasarkan inputan yang dilakukan pada proses sebelumnya, sistem mengubahnya menjadi bentuk pemrograman linear. Salah satunya adalah mengubah input menjadi fungsi kendala. Langkah-langkah proses membuat fungsi kendala dapat dilihat pada Gambar 4.6 berikut :



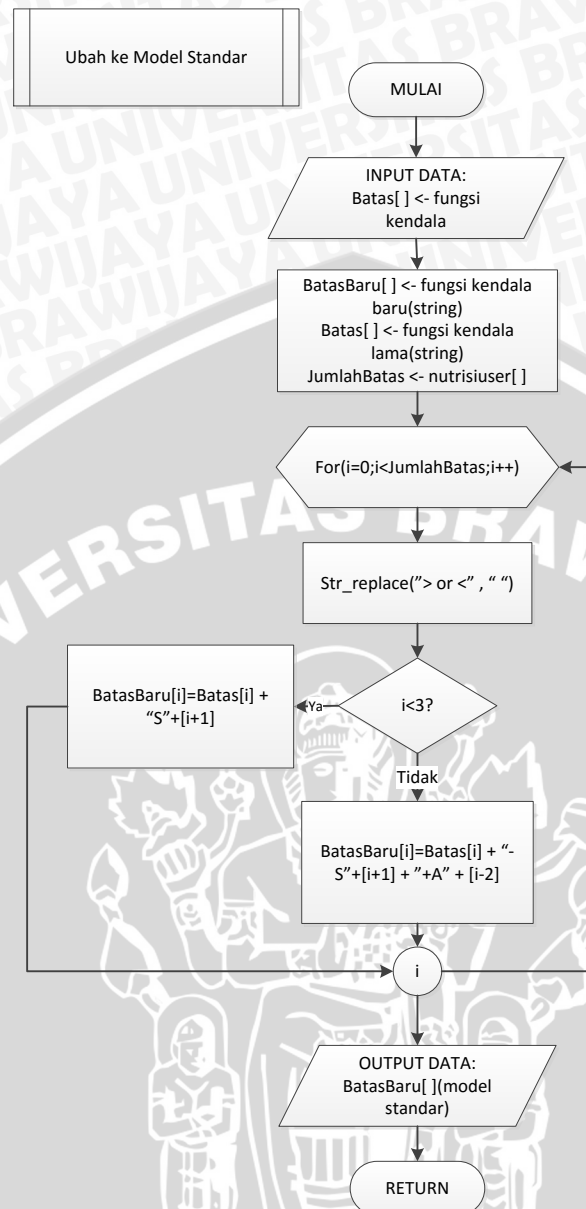
Gambar 4.6 Flowchart membuat fungsi kendala.

Berdasarkan Gambar 4.6 dapat dijelaskan langkah-langkah dalam membuat fungsi kendala adalah sebagai berikut :

1. Sistem akan mengubah dan memberi konstanta pada setiap makanan yang dipilih pada proses sebelumnya.
2. Sistem akan menampilkan bentuk dari fungsi kendala yang berupa pertidaksamaan dan batas kendalanya.

4.1.5 Ubah model standar

Pada proses ini adalah penambahan variabel surplus atau variabel *slack* atau variabel *artificial* ke dalam bentuk persamaan yang telah dilakukan pada proses 4.1.4. Proses penambahan variabel-variabel pembantu ini kemudian disebut model standar. Langkah-langkah proses untuk mengubah ke model standar dapat dilihat pada Gambar 4.7 berikut :



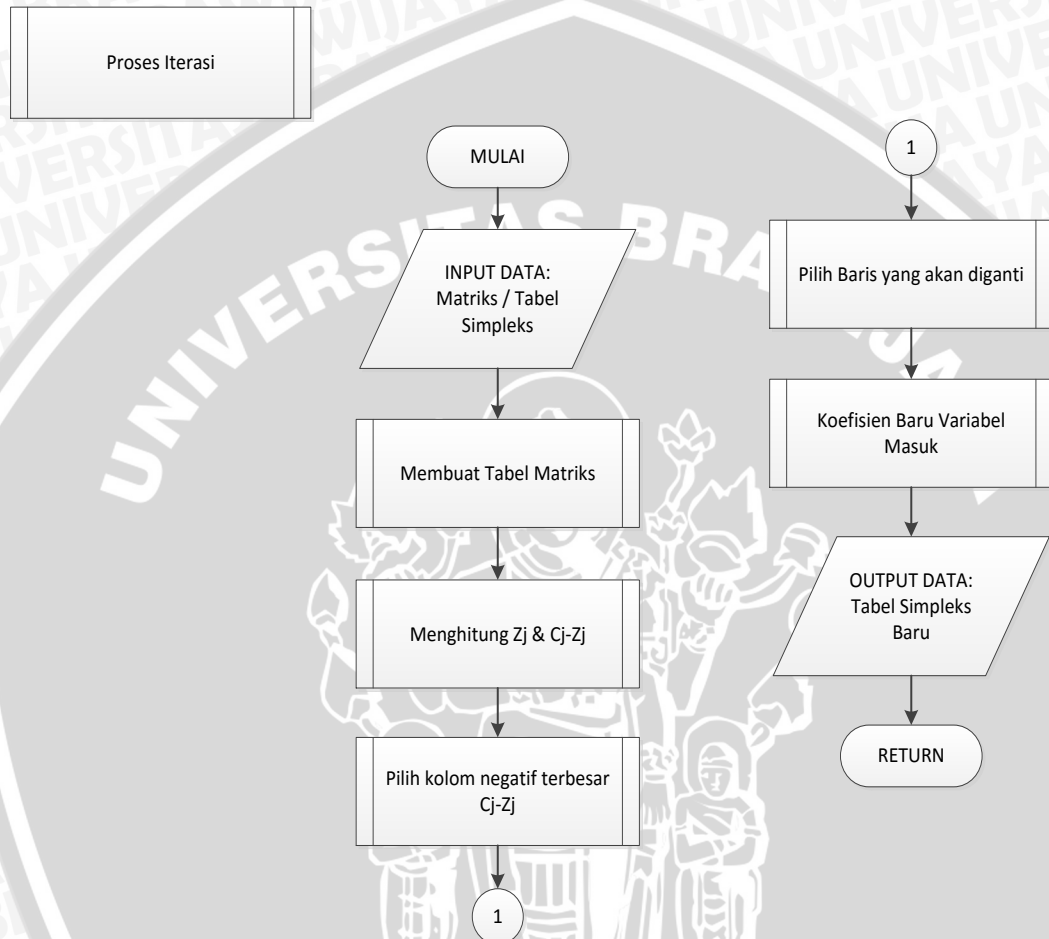
Gambar 4.7 Flowchart ubah ke model standar.

Berdasarkan Gambar 4.7 dapat dijelaskan langkah-langkah dalam mengubah ke model standar adalah sebagai berikut :

1. Sistem akan mengubah dan memberi konstanta berupa variabel surplus atau variabel slack atau variabel *artificial* pada kendala sesuai dengan bentuk batas dari kendala tersebut.
2. Sistem akan menampilkan bentuk dari fungsi kendala yang berupa persamaan beserta variabel tambahannya.

4.1.6 Proses iterasi

Pada proses ini menjelaskan langkah-langkah untuk menghitung permasalahan dengan metode Simpleks Big-M. Setelah memodelkan fungsi kendala ke model standar maka langkah selanjutnya adalah melakukan proses iterasi dengan metode Simpleks Big-M hingga ditemukan hasil akhir yang paling optimal. Langkah-langkah proses iterasi dapat dilihat pada Gambar 4.8 berikut :



Gambar 4.8 Flowchart proses iterasi.

Berdasarkan Gambar 4.8 dapat dijelaskan langkah-langkah dalam proses iterasi adalah sebagai berikut :

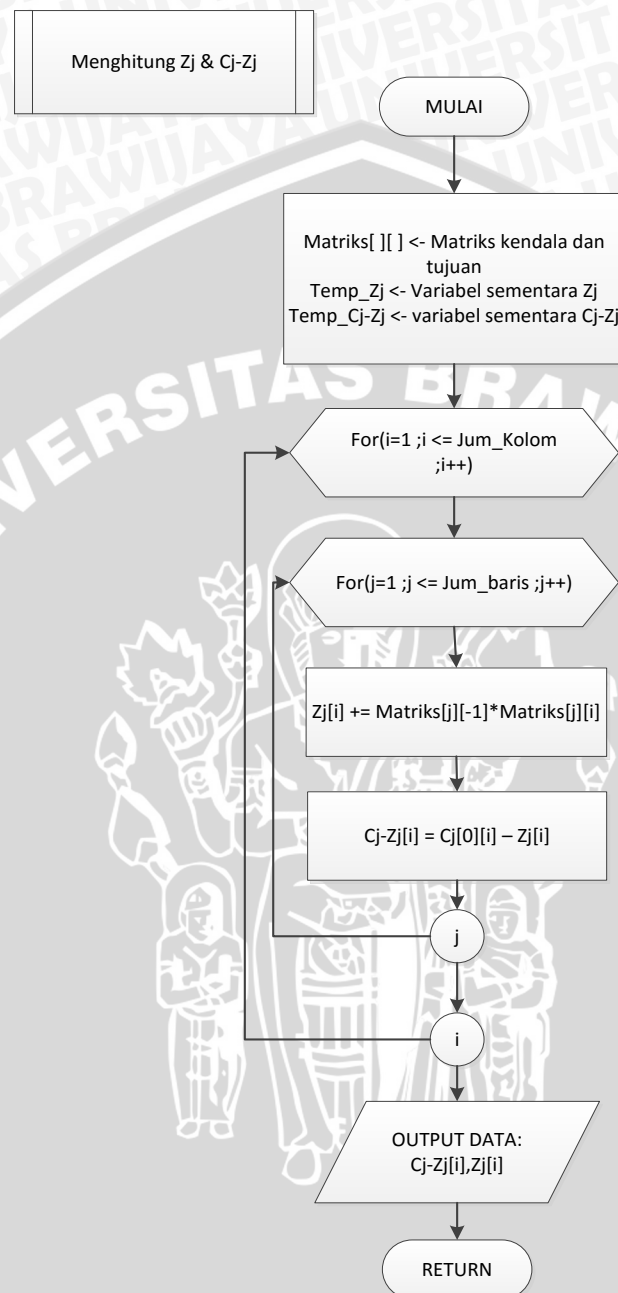
1. Sistem akan mengambil data dari tabel matriks.
2. Sistem akan membuat tabel matriks baru yang digunakan untuk melakukan perhitungan Simpleks Big-M.
3. Sistem menghitung nilai Z_j dan $C_j - Z_j$.
4. Sistem memilih kolom negatif terbesar $C_j - Z_j$ yang akan digunakan sebagai variabel yang akan masuk sebagai pengganti.
5. Sistem memilih baris yang akan diganti dengan berdasarkan kolom batas dibagi dengan kolom negatif terbesar $C_j - Z_j$ yang terpilih.



1. Sistem akan mengolah data dari inputan proses sebelumnya yaitu kendala-kendala dalam model standar dan fungsi tujuan.
2. Sistem akan memisahkan koefisien dengan variabel nya.
3. Sistem memasukkan koefisien ke dalam tabel matriks yang dibuat berdasarkan variabelnya. Variabel terlebih dahulu dipisah menjadi dua ,yaitu variabel basis dan variabel nonbasis dimana untuk variabel basis ,nilai awalnya sudah ditentukan.
4. Sistem menampilkan output berupa tabel Simpleks.

Pada proses ini menjelaskan bagaimana menghitung nilai Z_j dan $C_j - Z_j$ berdasarkan tabel matriks. Pada proses iterasi selanjutnya nilai $C_j - Z_j$ pada masing-

masing kolom juga mempengaruhi apakah hasil sudah optimal atau belum. Jika belum optimal maka akan dilakukan perhitungan ke iterasi berikutnya. Langkah-langkah menghitung Z_j dan $C_j - Z_j$ dapat dilihat pada Gambar 4.10 berikut :



Gambar 4.10 Flowchart menghitung Z_j dan $C_j - Z_j$.

Berdasarkan Gambar 4.10 dapat dijelaskan langkah-langkah dalam mengubah ke bentuk matriks adalah sebagai berikut :

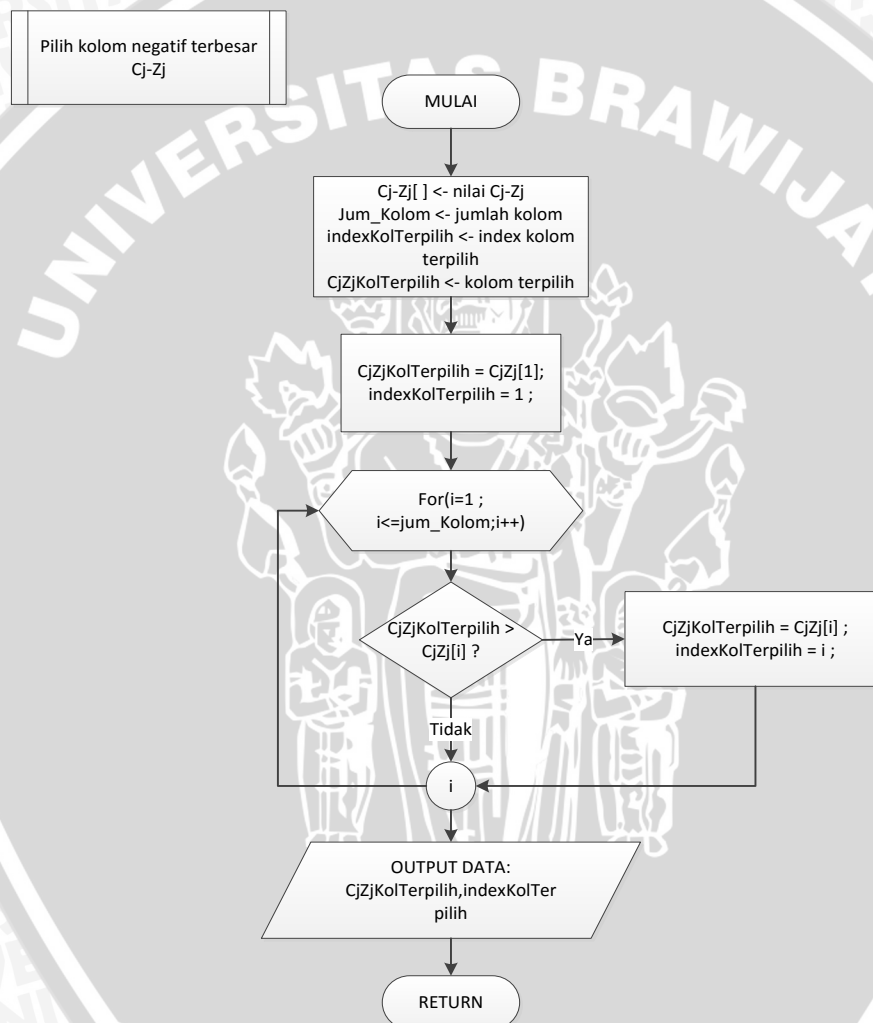
1. Sistem akan mengolah data dari tabel matriks yang sudah terbentuk.
2. Sistem akan melakukan perhitungan Z_j dan $C_j - Z_j$ pada setiap kolom variabel nonbasis maupun variabel basis. Nilai Z_j didapat dari $\sum$ perkalian nilai kolom

ke-m,baris ke-n dikalikan dengan kolom C_j baris ke-n . Sedangkan C_j didapat dari pengurangan nilai C_j kolom ke-n dengan Z_j kolom ke-n.

3. Sistem menampilkan output berupa nilai Z_j dan C_j-Z_j .

4.1.6.3 Pilih kolom negatif terbesar C_j-Z_j

Setelah menghitung nilai Z_j dan C_j-Z_j kemudian dilakukan proses selanjutnya yaitu menentukan kolom dengan nilai negatif C_j-Z_j terbesar. Kolom terpilih kemudian akan menjadi kolom terpilih yang akan masuk sebagai variabel dalam jawaban. Langkah-langkah memilih kolom negatif terbesar C_j-Z_j dapat dilihat pada Gambar 4.11 berikut :



Gambar 4.11 Flowchart pilih kolom negatif terbesar C_j-Z_j .

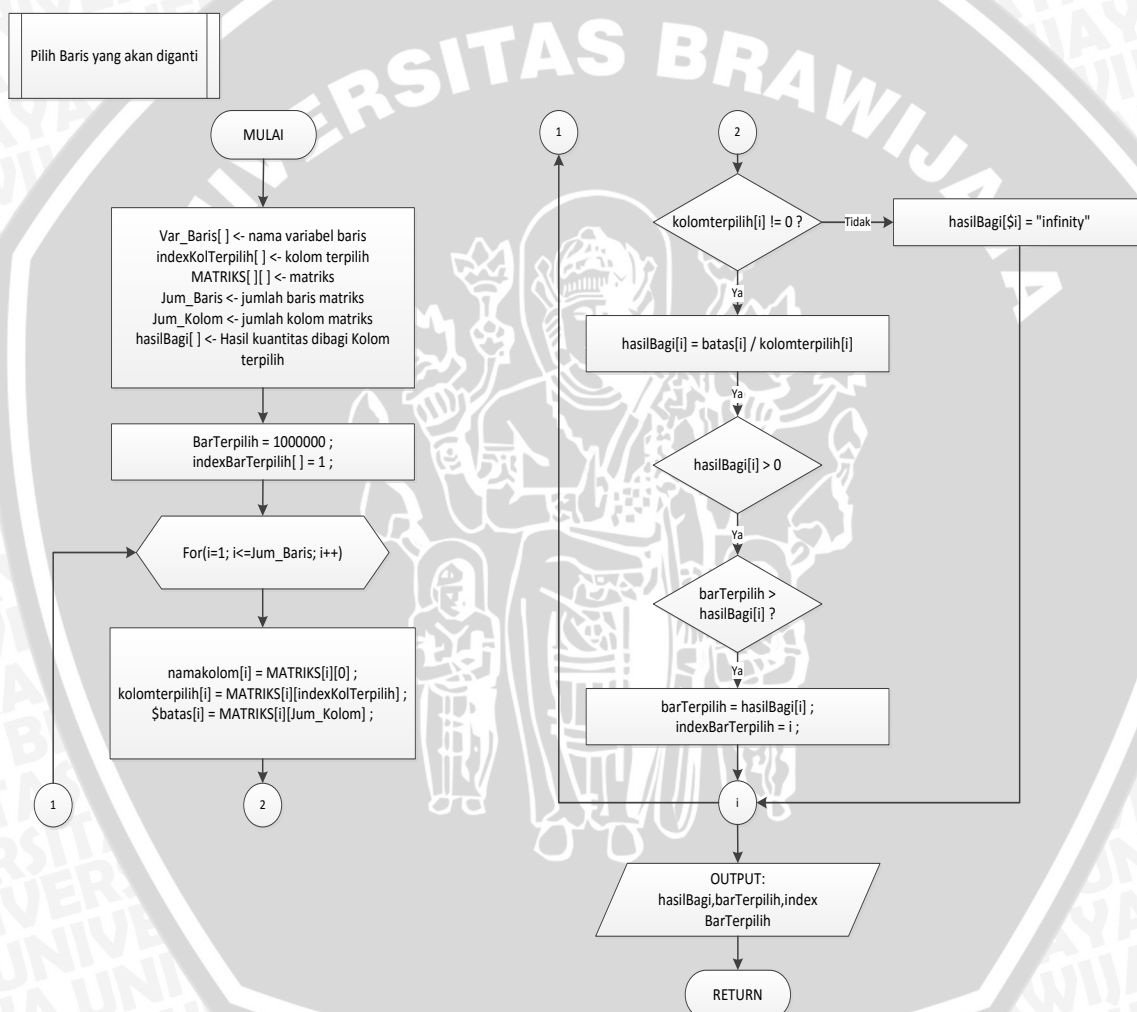
Berdasarkan Gambar 4.11 dapat dijelaskan langkah-langkah dalam mengubah ke bentuk matriks adalah sebagai berikut :

1. Sistem menerima data pembanding nilai C_j-Z_j dari masing-masing kolom.
2. Sistem akan melakukan pengecekan kolom mana sajakah yang memiliki nilai C_j-Z_j negatif.

3. Sistem kemudian membandingkan masing-masing kolom negatif C_j-Z_j terpilih, manakah diantaranya yang memiliki nilai terkecil (negatif terbesar).
4. Sistem menampilkan output berupa C_j-Z_j dengan nilai terkecil (negatif terbesar).

4.1.6.4 Pilih baris yang akan diganti

Pada proses ini adalah menentukan variabel dari baris yang akan diganti. Menentukan variabel yang akan diganti dilakukan dengan cara membagi nilai kuantitas dengan koefisien yang sebaris dengan kuantitas tersebut pada variabel yang masuk dalam jawaban. Variabel yang akan diganti ditentukan oleh hasil pembagian dengan nilai positif terkecil. Langkah-langkah memilih baris yang akan diganti dapat dilihat pada Gambar 4.12 berikut :



Gambar 4.12 Flowchart pilih baris yang akan diganti.

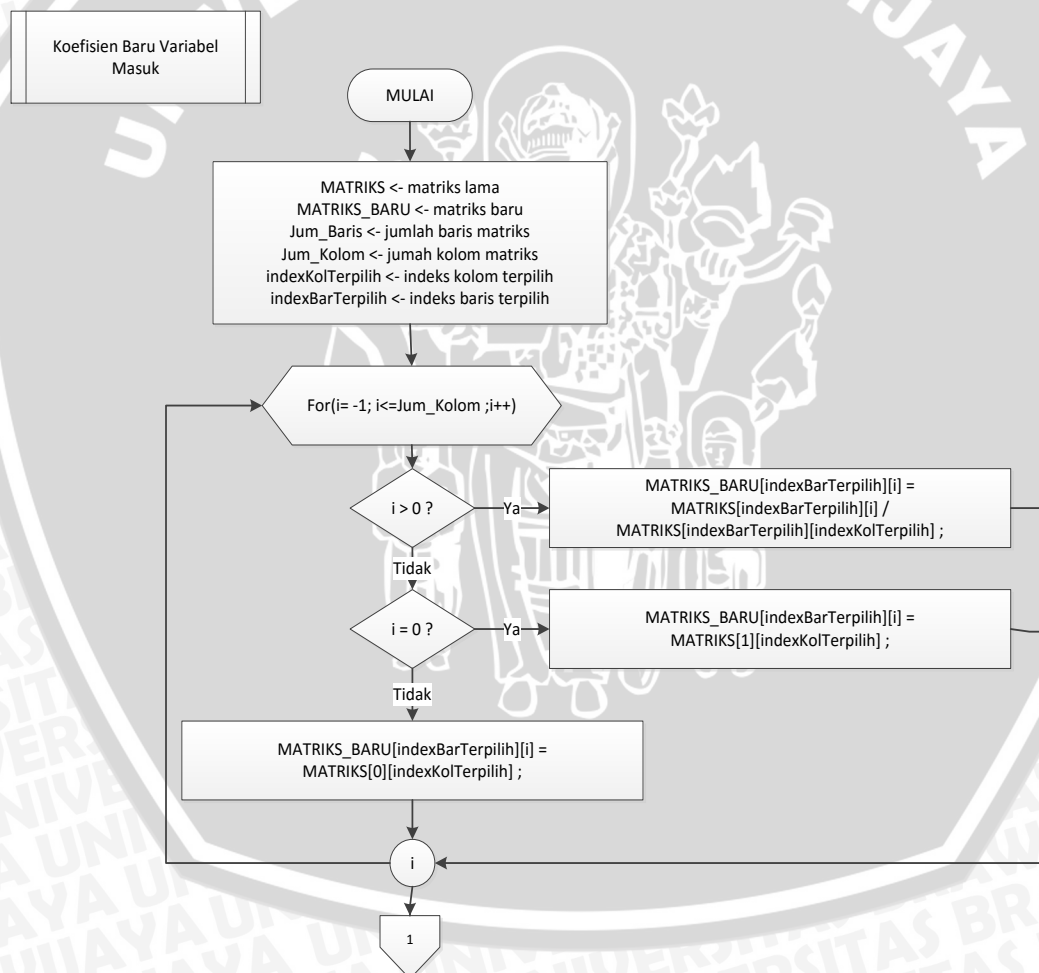
Berdasarkan Gambar 4.12 dapat dijelaskan langkah-langkah dalam memilih baris yang akan diganti adalah sebagai berikut :

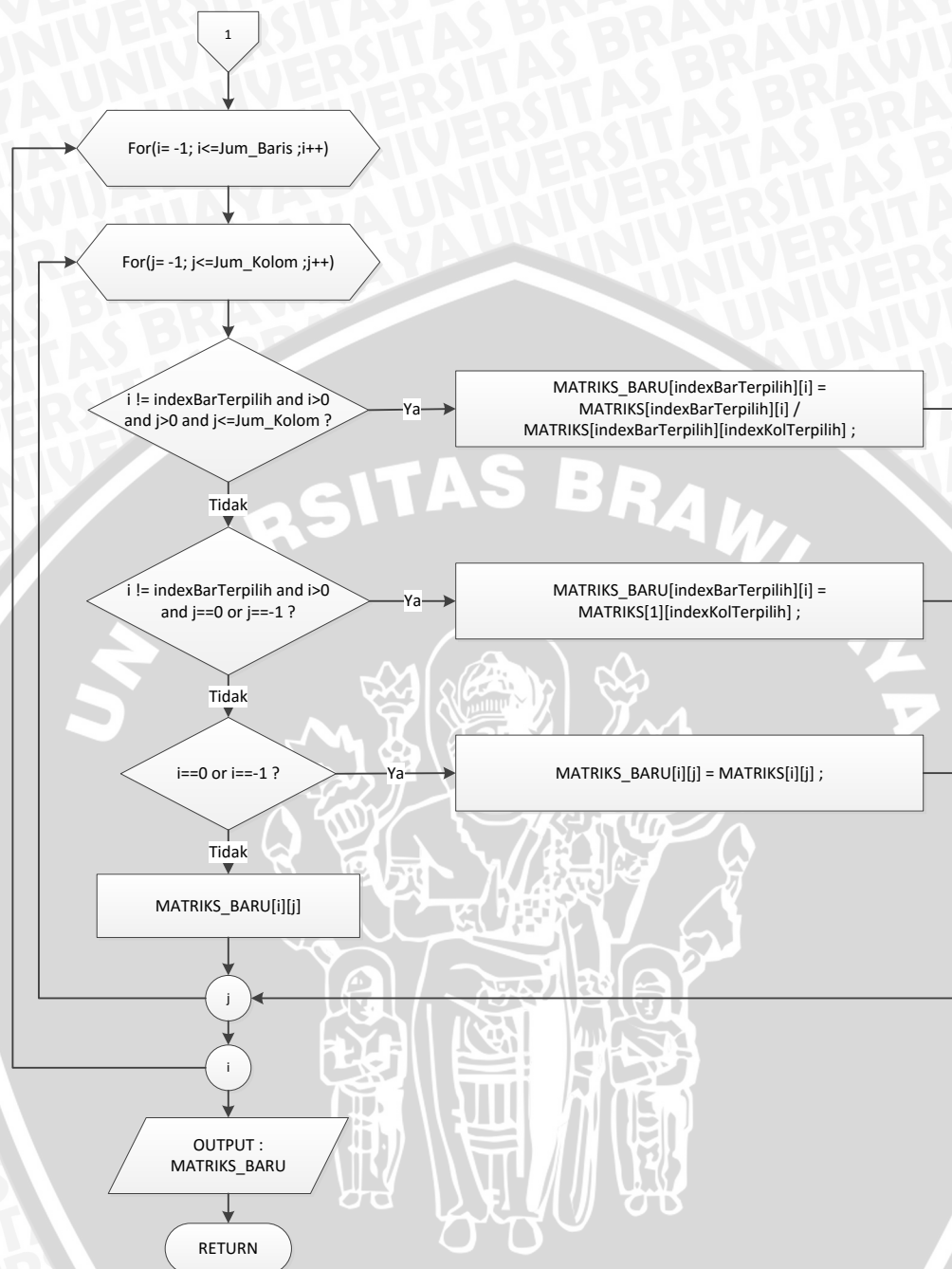
1. Sistem mengambil nilai kolom terpilih.

2. Sistem akan melakukan perhitungan tiap variabel baris dengan melakukan pembagian antara kolom kuantitas ke-m dengan kolom terpilih baris ke-m
3. Sistem melakukan pengecekan apakah nilai bagi variabel bernilai positif. Jika bernilai positif maka akan disimpan dalam variabel tertentu kemudian dibandingkan dengan nilai bagi variabel yang lain. Jika tidak maka akan melakukan perulangan lagi. Baris variabel yang akan diganti dipilih berdasarkan baris yang memiliki nilai bagi positif terkecil.
4. Sistem menampilkan output berupa baris dengan nilai positif terkecil.

4.1.6.5 Koefisien baru variabel masuk

Pada proses ini adalah menghitung nilai koefisien baru yang masuk pada variabel terpilih. Untuk variabel yang tidak terpilih kemudian juga direvisi nilainya dengan menggunakan nilai dari variabel terpilih yang telah direvisi sebelumnya. Langkah-langkah mengubah koefisien baru untuk variabel yang masuk dapat dilihat pada Gambar 4.13 berikut :





Gambar 4.13 Flowchart koefisien baru variabel masuk.

Berdasarkan Gambar 4.13 dapat dijelaskan langkah-langkah dalam membuat koefisien baru untuk variabel yang masuk adalah sebagai berikut :

1. Sistem mengambil nilai indeks kolom terpilih dan baris terpilih.
2. Sistem melakukan revisi pada variabel terpilih menggantikan variabel baris terpilih yang akan diganti.
3. Sistem melakukan revisi pada variabel lainnya dengan cara mengurangi nilai tiap baris ke-n, kolom ke-m baris lainnya dengan nilai perkalian antara nilai

perpotongan baris ke-n, kolom ke-m baris lainnya dengan nilai revisi variabel terpilih.

4. Sistem menampilkan output berupa tabel matriks yang baru.

4.2 Perhitungan manual

Perbedaan kebutuhan antar individu dan berbedanya nutrisi makanan yang dipilihnya menjadi permasalahan tersendiri bagi pemenuhan nutrisi setiap harinya. Pada subbab ini dipaparkan bagaimana proses untuk menentukan komposisi takaran makanan sesuai dengan menu yang dipilih sehingga mendapatkan total biaya yang optimum berdasarkan tingkat kebutuhan nutrisi setiap individu. Metode Simpleks Big-M diharapkan dapat menemukan solusi yang memenuhi batasan-batasan kebutuhan nutrisi individu dengan mendapatkan biaya yang optimal.

4.2.1 Menentukan kebutuhan nutrisi dan menu makanan

Tahap ini adalah tahap awal yang dibutuhkan untuk menentukan kebutuhan nutrisi dari individu sekaligus disesuaikan dengan menu makanan yang dipilih. Tahap ini juga merupakan tahap penginputan data yang kemudian diolah oleh proses-proses selanjutnya. Berikut contoh masukan :

- Nama Individu : Akbar Hakim Prabowo
- Jenis Kelamin : Laki-Laki
- Usia : 22 tahun
- Berat Badan : 65 kg
- Tinggi badan : 170 cm
- Jenis Aktivitas : Sedentary

Dari masukan tersebut, Kemudian dihitung *BMR* (*Basal Metabolic Rate*) yang dibutuhkan dengan menggunakan Harris Benedict Equation (2.2). Setelah menghitung *BMR* langkah selanjutnya adalah menentukan jumlah energi/kalori yang dikeluarkan dimana kemudian disebut *TEE* (*Total Energy Expenditure*).

Dengan menggunakan persamaan Harris Benedict (2.2) maka ditemukan hasil dari *BMR* individu :

- $BMR = 88.362 + (13.397 \times 65) + (4.799 \times 170) - (5.677 \times 22)$
 $BMR = 1650,103$
- $TEE = 1558,132 \times 1.2$
 $TEE = 1980,1236 \text{ kalori}$

Setelah menghitung *TEE* maka langkah selanjutnya adalah menghitung nutrisi yang dibutuhkan meliputi protein, lemak dan karbohidrat (*macronutrient*) yang memiliki nilai kebutuhan yang berbeda-beda. Nilai *macronutrient* ini berupa rentang dari batas bawah dan batas atas. Lalu dari kebutuhan yang masih bernilai

kalori ini dikonversi dalam bentuk gram. Untuk mengkonversi ke dalam gram, protein dan karbohidrat harus dibagi dengan 4 sedangkan untuk lemak dibagi dengan 9.

- Protein batas bawah(10%) :
 $= 10/100 \times 1980,1236 = 198,01236$ kalori
 $= 198,01236 / 4 = 49,50309$ gram
- Protein batas atas(35%) :
 $= 35/100 \times 1980,1236 = 693,04326$ kalori
 $= 693,04326 / 4 = 173,260815$ gram
- Lemak batas bawah(20%) :
 $= 20/100 \times 1980,1236 = 396,02472$ kalori
 $= 396,02472 / 9 = 44,00274667$ gram
- Lemak batas atas(35%) :
 $= 35/100 \times 1980,1236 = 693,04326$ kalori
 $= 693,04326 / 9 = 77,00480667$ gram
- Karbohidrat batas bawah(45%) :
 $= 45/100 \times 1980,1236 = 891,05562$ kalori
 $= 891,05562 / 4 = 222,763905$ gram
- Karbohidrat batas atas(65%) :
 $= 65/100 \times 1980,1236 = 1287,08034$ kalori
 $= 1287,08034 / 4 = 321,770085$ gram

Setelah kebutuhan individu telah ditentukan, individu memilih makanan yang akan dikonsumsi selama sehari. Sebagai contoh makanan yang dikonsumsi beserta harga dan nutrisinya dilihat pada Tabel 4.1 :

Tabel 4.1 Tabel makanan beserta nutrisi dan harga

Makanan	Kalori	Protein (/150gr)	Lemak (/150gr)	Karbohidrat (/150gr)	Harga (/150gr)
tempe	193	18,54	10,8	9,39	850
mangga	107	0,84	0,45	28,05	1850
melon	19	0,46	0,1	4,49	2250

4.2.2 Mengubah ke bentuk *pemrograman linear*

Pada tahap ini yang perlu dilakukan adalah menentukan fungsi tujuan dan fungsi kendalanya lalu memodelkannya dalam bentuk pertidaksamaan (*pemrograman linear*). Pada permasalahan ini maka fungsi tujuannya adalah harga sedangkan fungsi kendalanya adalah protein, lemak dan karbohidrat (*macronutrient*) dari masing-masing makanan dimana batasannya adalah batas atas dan batas bawah dari kebutuhan macronutrient individu.

$$Z = 850 \quad x_1 + 1850x_2 + 2250x_3$$

s.k

- $18,54x_1 + 0,84x_2 + 0,46x_3 \leq 173,260815$
- $10,8x_1 + 0,45x_2 + 0,1x_3 \leq 77,00480667$
- $9,39x_1 + 28,05x_2 + 4,49x_3 \leq 321,770085$
- $18,54x_1 + 0,84x_2 + 0,46x_3 \geq 49,50309$
- $10,8x_1 + 0,45x_2 + 0,1x_3 \geq 44,00274667$
- $9,39x_1 + 28,05x_2 + 4,49x_3 \geq 222,763905$
- $193x_1 + 107x_2 + 19x_3 = 1980,1236$
- $x_1 \geq 1$
- $x_2 \geq 1$
- $x_3 \geq 1$

4.2.3 Mengubah ke model standar

Setelah fungsi tujuan dan fungsi kendala diketahui maka selanjutnya dilakukan proses pengubahan ke bentuk standar, yaitu dengan cara menambahkan variabel *surplus* atau variabel *slack* atau variabel *artificial* sehingga bentuk pertidaksamaan sebelumnya akan berubah ke bentuk persamaan seperti dibawah ini :

$$Z = 850 \quad x_1 + 1850x_2 + 2250x_3 + M(A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7)$$

s.k

- $18,54x_1 + 0,84x_2 + 0,46x_3 + S_1 \leq 173,260815$
- $10,8x_1 + 0,45x_2 + 0,1x_3 + S_2 \leq 77,00480667$
- $9,39x_1 + 28,05x_2 + 4,49x_3 + S_3 \leq 321,770085$
- $18,54x_1 + 0,84x_2 + 0,46x_3 - S_4 + A_1 \geq 49,50309$
- $10,8x_1 + 0,45x_2 + 0,1x_3 - S_5 + A_2 \geq 44,00274667$
- $9,39x_1 + 28,05x_2 + 4,49x_3 - S_6 + A_3 \geq 222,763905$
- $193x_1 + 107x_2 + 19x_3 + A_4 = 1980,1236$
- $x_1 - S_7 + A_5 \geq 1$
- $x_2 - S_8 + A_6 \geq 1$
- $x_3 - S_9 + A_7 \geq 1$

4.2.4 Mengubah ke tabel matriks

Setelah melakukan pemodelan standar yaitu dengan memberikan Variabel *surplus*/Variabel *Artificial* , langkah selanjutnya yang perlu dilakukan adalah melakukan penyelesaian permasalahan dengan metode Simpleks Big-M. Adapun langkah-langkah penerapan metode Simpleks Big-M dapat dilihat pada 2.7.3.

Tabel 4.2 Tabel awal Simpleks

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	18,54	0,84	0,46	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	173,26082
0	S2	10,8	0,45	0,1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77,004807
0	S3	9,39	28,05	4,49	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	321,77009
10000	A1	18,54	0,84	0,46	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	49,50309
10000	A2	10,8	0,45	0,1	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	44,002747
10000	A3	9,39	28,05	4,49	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	222,76391
10000	A4	193	107	19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1980,1236
10000	A5	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1
10000	A6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	2327300	1373400	250500	0	0	0	-10000	10000	-10000	10000	-10000	10000	10000	-10000	10000	-10000	10000	-10000	10000	22993933
	Cj-Zj	-2325450	-1371150	-249650	0	0	0	10000	0	10000	0	10000	0	0	10000	0	10000	0	10000	0	

Nilai Zj didapat dari $\sum$ baris Xn x baris Kolom Cj. Sedangkan Cj-Zj didapat dari nilai Cj dikurangkan dengan nilai Zj. Setelah mendapatkan nilai Cj-Zj, kemudian menentukan Variabel yang masuk dalam jawaban dengan memilih Cj-Zj dengan nilai negatif yang paling minimum. Dari Tabel maka didapatkan nilai negatif minimum adalah -2325450 untuk variabel x1. Maka x1 dipilih sebagai variabel yang masuk dalam jawaban/Kolom kunci. Langkah berikutnya menentukan variabel yang akan diganti. Cara untuk menentukan variabel yang akan diganti adalah dengan menentukan nilai Rasio solusi positif terkecil dari tiap-tiap baris. Adapun perhitungan rasio solusi adalah sebagai berikut.

Rasio = nilai batas / nilai kolom terpilih

(4.1)

Sehingga dari rumus tersebut didapatkan :

Tabel 4.3 Tabel menentukan variabel yang akan diganti

kolom	x1	batas	batas/x1
S1	18,54	173,2608	9,34524353
S2	10,8	77,00481	7,13007469
S3	9,39	321,7701	34,2673147
A1	18,54	49,50309	2,67006958
A2	10,8	44,00275	4,0743284
A3	9,39	222,7639	23,7235256
A4	193	1980,124	10,2597078
A5	1	1	1
A6	0	1	#DIV/0!
A7	0	1	#DIV/0!

Dari perhitungan pada Tabel 4.4 dapat diketahui bahwa variabel yang akan diganti/Baris kunci adalah **baris A5** (yaitu 1). Setelah kolom kunci dan baris kunci didapatkan, maka didapatkan juga nilai perpotongan antara baris dan kolom kunci tersebut. Dari Tabel 4.3 dan Tabel 4.4 maka didapatkan nilai perpotongan antara kolom x5 dan baris A9 adalah 1. Langkah selanjutnya dilakukan perhitungan koefisien baru untuk variabel yang masuk dengan rumus :

Baris kunci baru = Baris kunci lama / nilai perpotongan

(4.2)

Sehingga dari rumus tersebut didapatkan :

Tabel 4.4 Menghitung koefisien baru untuk variabel yang masuk

A5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1
x1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1

Setelah didapatkan nilai koefisien pada variabel baru yang masuk , langkah berikutnya adalah melakukan revisi koefisien terhadap baris lainnya. Adapun rumus yang digunakan adalah sebagai berikut.

$$\text{Baris baru} = \text{Baris lama} - (\text{nilai perpotongan kolom kunci} \times \text{baris kunci baru}) \quad (4.3)$$

Sehingga dari rumus tersebut didapatkan :

Tabel 4.5 Revisi baris lain

S1	18,54	0,84	0,46	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	173,26082	
18,54	0	0,84	0,46	1	0	0	0	0	0	0	0	0	0	0	18,54	-18,54	0	0	0	0	154,72082
S2	10,8	0,45	0,1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77,004807	
10,8	0	0,45	0,1	0	1	0	0	0	0	0	0	0	0	0	10,8	-10,8	0	0	0	0	66,204807
S3	9,39	28,05	4,49	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	321,77009	
9,39	0	28,05	4,49	0	0	1	0	0	0	0	0	0	0	0	9,39	-9,39	0	0	0	0	312,38009
A1	18,54	0,84	0,46	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	49,50309	
18,54	0	0,84	0,46	0	0	0	-1	1	0	0	0	0	0	0	18,54	-18,54	0	0	0	0	30,96309
A2	10,8	0,45	0,1	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	44,002747	
10,8	0	0,45	0,1	0	0	0	0	0	-1	1	0	0	0	0	10,8	-10,8	0	0	0	0	33,202747
A3	9,39	28,05	4,49	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	222,76391	
9,39	0	28,05	4,49	0	0	0	0	0	0	0	-1	1	0	0	9,39	-9,39	0	0	0	0	213,37391
A4	193	107	19	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1980,1236	
193	0	107	19	0	0	0	0	0	0	0	0	0	0	1	193	-193	0	0	0	0	1787,1236
A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
A6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1

Dari seluruh perhitungan di atas maka nilai-nilai baris baru yang sudah didapatkan kemudian dapat dituangkan ke dalam bentuk Tabel Simpleks ke II. Seperti pada Tabel 4.6 berikut ini.

Tabel 4.6 Tabel Simpleks II

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0,84	0,46	1	0	0	0	0	0	0	0	0	0	18,54	-18,54	0	0	0	0	154,72082
0	S2	0	0,45	0,1	0	1	0	0	0	0	0	0	0	0	10,8	-10,8	0	0	0	0	66,204807
0	S3	0	28,05	4,49	0	0	1	0	0	0	0	0	0	0	9,39	-9,39	0	0	0	0	312,38009
10000	A1	0	0,84	0,46	0	0	0	-1	1	0	0	0	0	0	18,54	-18,54	0	0	0	0	30,96309
10000	A2	0	0,45	0,1	0	0	0	0	0	-1	1	0	0	0	10,8	-10,8	0	0	0	0	33,202747
10000	A3	0	28,05	4,49	0	0	0	0	0	0	0	-1	1	0	9,39	-9,39	0	0	0	0	213,37391
10000	A4	0	107	19	0	0	0	0	0	0	0	0	0	1	193	-193	0	0	0	0	1787,1236
850	x1	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	1
10000	A6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1373400	250500	0	0	0	-10000	10000	-10000	10000	-10000	10000	10000	2316450	-2316450	-10000	10000	-10000	10000	20667483
	Cj-Zj	0	-1371550	-248250	0	0	0	10000	0	10000	0	10000	0	0	-2316450	2326450	10000	0	10000	0	

Pada Tabel 4.6 dapat dikatakan masih belum optimal ,karena pada baris Cj-Zj masih terdapat nilai yang bernilai negatif. Untuk itu untuk memperbaiki hasil diperlukan iterasi dengan langkah yang sama seperti langkah-langkah sebelumnya. Iterasi akan terus dilakukan hingga mendapatkan hasil yang paling optimal(tidak ada Cj-Zj yang bernilai negatif). Dan berturut-turut Tabel yang terbentuk setelah dilakukannya proses iterasi adalah sebagai berikut.

Tabel 4.7 Tabel Simpleks III

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0	1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	123,75773
0	S2	0	-0,0393204	-0,167961	0	1	0	0,58252	-0,5825	0	0	0	0	0	0	0	0	0	0	0	48,168055
0	S3	0	27,6245631	4,2570227	0	0	1	0,50647	-0,5065	0	0	0	0	0	0	0	0	0	0	0	296,69813
0	S7	0	0,04530744	0,0248112	0	0	0	-0,05394	0,05394	0	0	0	0	0	1	-1	0	0	0	0	1,6700696
10000	A2	0	-0,0393204	-0,167961	0	0	0	0,58252	-0,5825	-1	1	0	0	0	0	0	0	0	0	0	15,165995
10000	A3	0	27,6245631	4,2570227	0	0	0	0,50647	-0,5065	0	0	-1	1	0	0	0	0	0	0	0	197,69195
10000	A4	0	98,2556634	14,211435	0	0	0	10,4099	-10,41	0	0	0	0	1	0	0	0	0	0	0	1464,8002
850	x1	1	0,04530744	0,0248112	0	0	0	-0,05394	0,05394	0	0	0	0	0	0	0	0	0	0	0	2,6700696
10000	A6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1268447,57	193026,05	0	0	0	114943	-114943	-10000	10000	-10000	10000	10000	0	0	-10000	10000	-10000	10000	16798851
	Cj-Zj	0	-1266597,6	-190776,1	0	0	0	-114943	124943	10000	0	10000	0	0	0	10000	10000	0	10000	0	

Tabel 4.8 Tabel Simpleks IV

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0	1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	123,75773
0	S2	0	0	-0,167961	0	1	0	0,58252	-0,5825	0	0	0	0	0	0	0	-0,039320388	0,0393	0	0	48,207376
0	S3	0	0	4,2570227	0	0	1	0,50647	-0,5065	0	0	0	0	0	0	0	27,62456311	-27,62	0	0	269,07357
0	S7	0	0	0,0248112	0	0	0	-0,05394	0,05394	0	0	0	0	0	1	-1	0,045307443	-0,045	0	0	1,6247621
10000	A2	0	0	-0,167961	0	0	0	0,58252	-0,5825	-1	1	0	0	0	0	0	-0,039320388	0,0393	0	0	15,205316
10000	A3	0	0	4,2570227	0	0	0	0,50647	-0,5065	0	0	-1	1	0	0	0	27,62456311	-27,62	0	0	170,06739
10000	A4	0	0	14,211435	0	0	0	10,4099	-10,41	0	0	0	0	1	0	0	98,25566343	-98,26	0	0	1366,5445
850	x1	1	0	0,0248112	0	0	0	-0,05394	0,05394	0	0	0	0	0	0	0	0,045307443	-0,045	0	0	2,6247621
1850	X2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	193026,05	0	0	0	114943	-114943	-10000	10000	-10000	10000	10000	0	0	1256597,573	-1E+06	-10000	10000	15532253
	Cj-Zj	0	0	-190776,1	0	0	0	-114943	124943	10000	0	10000	0	0	0	10000	-1256597,573	1E+06	10000	0	

Tabel 4.9 Tabel Simpleks V

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0	1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	123,75773
0	S2	0	0	-0,161902	0	1	0	0,58325	-0,5832	0	0	-0,001	0,0014	0	0	0	0	0	0	0	48,449447
0	S3	0	0	0	0	0	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	99,00618
0	S7	0	0	0,0178292	0	0	0	-0,05477	0,05477	0	0	0,0016	-0,002	0	1	-1	0	0	0	0	1,3458322
10000	A2	0	0	-0,161902	0	0	0	0,58325	-0,5832	-1	1	-0,001	0,0014	0	0	0	0	0	0	0	15,447387
0	S8	0	0	0,1541028	0	0	0	0,01833	-0,0183	0	0	-0,036	0,0362	0	0	0	1	-1	0	0	6,1563829
10000	A4	0	0	-0,930039	0	0	0	8,60849	-8,6085	0	0	3,5568	-3,557	1	0	0	0	0	0	0	761,64502
850	x1	1	0	0,0178292	0	0	0	-0,05477	0,05477	0	0	0,0016	-0,002	0	0	0	0	0	0	0	2,3458322
1850	x2	0	1	0,1541028	0	0	0	0,01833	-0,0183	0	0	-0,036	0,0362	0	0	0	0	0	0	0	7,1563829
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	-619,1586	0	0	0	91904,7	-91905	-10000	10000	35488	-35488	10000	0	0	0	0	-10000	10000	7796157,3
	Cj-Zj	0	0	2869,1586	0	0	0	-91904,7	101905	10000	0	-35488	45488	0	0	10000	0	10000	10000	0	

Tabel 4.10 Tabel Simpleks VI

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0,2775879	1	0	0	0	0	1,71454	-1,7145	0,0024	-0,002	0	0	0	0	0	0	0	97,272488
0	S2	0	0	0	0	1	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	33,00206
0	S3	0	0	0	0	0	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	99,00618
0	S7	0	0	0,0026263	0	0	0	0	0	-0,0939	0,0939	0,0015	-0,002	0	1	-1	0	0	0	0	2,7963784
0	S4	0	0	-0,277588	0	0	0	1	-1	-1,7145	1,71454	-0,002	0,0024	0	0	0	0	0	0	0	26,485237
0	S8	0	0	0,1591921	0	0	0	0	0	0,03143	-0,0314	-0,036	0,0362	0	0	0	1	-1	0	0	5,670799
10000	A4	0	0	1,4595743	0	0	0	0	0	14,7596	-14,76	3,5778	-3,578	1	0	0	0	0	0	0	533,64707
850	x1	1	0	0,0026263	0	0	0	0	0	-0,0939	0,0939	0,0015	-0,002	0	0	0	0	0	0	0	3,7963784
1850	x2	0	1	0,1591921	0	0	0	0	0	0,03143	-0,0314	-0,036	0,0362	0	0	0	0	0	0	0	6,670799
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	24892,481	0	0	0	0	0	147575	-147575	35713	-35713	10000	0	0	0	0	-10000	10000	5362038,6
	Cj-Zj	0	0	-22642,48	0	0	0	0	10000	-147575	157575	-35713	45713	0	0	10000	0	10000	10000	0	

Tabel 4.11 Tabel Simpleks VII

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0,2775879	1	-1,7145	0	0	0	0	0	0,0024	-0,002	0	0	0	0	0	0	0	40,688976
0	S5	0	0	0	0	1	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	33,00206
0	S3	0	0	0	0	0	1	0	0	0	0	1	-1	0	0	0	0	0	0	0	99,00618
0	S7	0	0	0,0026263	0	0,0939	0	0	0	0	0	0,0015	-0,002	0	1	-1	0	0	0	0	5,8953501
0	S4	0	0	-0,277588	0	1,71454	0	1	-1	0	0	-0,002	0,0024	0	0	0	0	0	0	0	83,068749
0	S8	0	0	0,1591921	0	-0,0314	0	0	0	0	0	-0,036	0,0362	0	0	0	1	-1	0	0	4,6333892
10000	A4	0	0	1,4595743	0	-14,76	0	0	0	0	0	3,5778	-3,578	1	0	0	0	0	0	0	46,548378
850	x1	1	0	0,0026263	0	0,0939	0	0	0	0	0	0,0015	-0,002	0	0	0	0	0	0	0	6,8953501
1850	x2	0	1	0,1591921	0	-0,0314	0	0	0	0	0	-0,036	0,0362	0	0	0	0	0	0	0	5,6333892
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	24892,481	0	-147575	0	0	0	0	0	35713	-35713	10000	0	0	0	0	-10000	10000	491766,6
	Cj-Zj	0	0	-22642,48	0	147575	0	0	10000	0	10000	-35713	45713	0	0	10000	0	10000	10000	0	

Tabel 4.12 Tabel Simpleks VIII

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0,2765923	1	-1,7045	0	0	0	0	0	0	0	-0	0	0	0	0	0	0	40,657226
0	S5	0	0	0	0	1	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	33,00206
0	S3	0	0	-0,407949	0	4,12531	1	0	0	0	0	0	0	-0,28	0	0	0	0	0	0	85,995957
0	S7	0	0	0,0020117	0	0,10012	0	0	0	0	0	0	0	-0	1	-1	0	0	0	0	5,8757508
0	S4	0	0	-0,276592	0	1,70448	0	1	-1	0	0	0	0	7E-04	0	0	0	0	0	0	83,100499
0	S8	0	0	0,1739415	0	-0,1806	0	0	0	0	0	0	0	0,01	0	0	1	-1	0	0	5,1037728
0	S6	0	0	0,4079495	0	-4,1253	0	0	0	0	0	1	-1	0,279	0	0	0	0	0	0	13,010223
850	x1	1	0	0,0020117	0	0,10012	0	0	0	0	0	0	0	-0	0	0	0	0	0	0	6,8757508
1850	x2	0	1	0,1739415	0	-0,1806	0	0	0	0	0	0	0	0,01	0	0	0	0	0	0	6,1037728
10000	A7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	10323,502	0	-248,98	0	0	0	0	0	0	0	18,34	0	0	0	0	-10000	10000	27136,368
	Cj-Zj	0	0	-8073,502	0	248,982	0	0	10000	0	10000	0	10000	9982	0	10000	0	10000	10000	0	

Tabel 4.13 Tabel Simpleks IX

Kolom Cj	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7	BATAS
	Cj	850	1850	2250	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000	
0	S1	0	0	0	1	-1,7045	0	0	0	0	0	0	0	-0	0	0	0	0	0,27659	-0,27659	40,380633
0	S2	0	0	0	0	1	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	33,00206
0	S3	0	0	0	0	4,12531	1	0	0	0	0	0	0	-0,28	0	0	0	0	-0,40795	0,40795	86,403907
0	S7	0	0	0	0	0,10012	0	0	0	0	0	0	0	-0	1	-1	0	0	0,00201	-0,00201	5,8737391
0	S4	0	0	0	0	1,70448	0	1	-1	0	0	0	0	7E-04	0	0	0	0	-0,27659	0,27659	83,377092
0	S8	0	0	0	0	-0,1806	0	0	0	0	0	0	0	0,01	0	0	1	-1	0,17394	-0,17394	4,9298313
0	S6	0	0	0	0	-4,1253	0	0	0	0	0	1	-1	0,279	0	0	0	0	0,40795	-0,40795	12,602273
850	x1	1	0	0	0	0,10012	0	0	0	0	0	0	0	-0	0	0	0	0	0,00201	-0,00201	6,8737391
1850	x2	0	1	0	0	-0,1806	0	0	0	0	0	0	0	0,01	0	0	0	0	0,17394	-0,17394	5,9298313
2250	x3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	1
	Zj	850	1850	2250	0	-248,98	0	0	0	0	0	0	0	18,34	0	0	0	0	-1926,5	1926,5	19062,866
	Cj-Zj	0	0	0	0	248,982	0	0	10000	0	10000	0	10000	9982	0	10000	0	10000	1926,5	8073,5	

Pada Tabel 4.13 dapat dilihat bahwa pada baris Cj-Zj sudah tidak ditemukan lagi kolom yang bernilai negatif. Dari hasil ini kemudian dapat disimpulkan bahwa pada iterasi ke-8 ,iterasi sudah menampilkan hasil yang paling optimal. Dengan begitu iterasi sudah tidak perlu dilakukan kembali.

Setelah proses iterasi selesai maka selanjutnya adalah membaca Tabel Simpleks yang sudah optimal. Jika dilihat dari Tabel 4.13 tersebut dapat dilihat bahwa nilai x1(tempe), x2(mangga) dan x3(melon) secara berturut-turut adalah 6,874 ; 5,93 dan 1 satuan. Dan Jumlah uang yang harus dibayarkan adalah 19062,866 rupiah (per 150gr). Untuk mengkonversi berapa gram yang dibutuhkan oleh individu maka didapatkan dengan perhitungan berikut.

- $X1 \text{ gram} / 150 \text{ gram} = 6,874$
 $X1 = 6,874 \times 150$
 $X1 = 1031,1 \text{ gram}$

- $X2 \text{ gram} / 150 \text{ gram} = 5,93$
 $X2 = 5,93 \times 150$
 $X2 = 889,5 \text{ gram}$
- $X3 \text{ gram} / 150 \text{ gram} = 1$
 $X3 = 1 \times 150$
 $X3 = 150 \text{ gram}$

Dari perhitungan diatas maka didapatkan jumlah gram yang harus dikonsumsi oleh individu berdasarkan makanan yang diinginkan beserta jumlah biaya paling minimum yang harus dikeluarkan oleh individu dalam sehari. Dari perhitungan tersebut didapatkan individu harus mengkonsumsi :

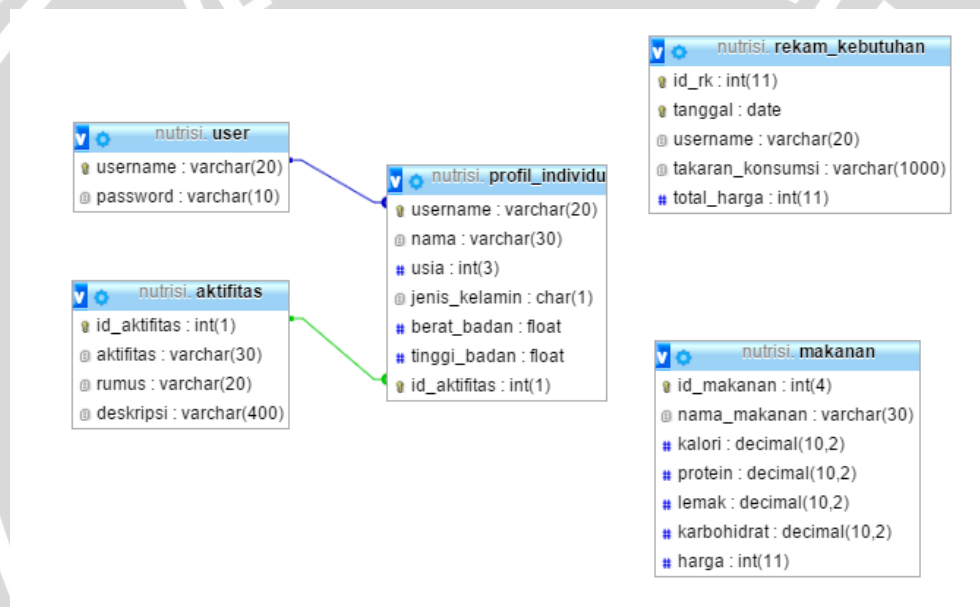
- Tempe seberat 1031,1 gram
- Mangga seberat 889,5 gram
- Melon seberat 150 gram

Dengan total biaya yang dikeluarkan adalah 19062,866 rupiah.

4.3 Perancangan database

Database digunakan untuk menampung data-data yang dibutuhkan seperti data individu, makanan dan rekam kebutuhan dari individu. Terdapat lima Tabel dalam database nutrisi yaitu user, aktifitas, profil_individu, makanan dan rekam_kebutuhan.

Tabel user terdiri dari *field* *username* dan *password*. Tabel aktifitas terdiri dari *field* *id_aktifitas*, *aktifitas*, *rumus* dan *deskripsi*. Tabel profil_individu terdiri dari *field* *username*, *nama*, *usia*, *jenis_kelamin*, *berat_badan*, *tinggi_badan* dan *id_aktifitas*. Tabel makanan terdiri dari *field* *id_makanan*, *nama_makanan*, *protein*, *lemak*, *karbohidrat* dan *harga*. Tabel rekam_kebutuhan terdiri dari *field* *id_rk*, *tanggal*, *username*, *takaran_konsumsi* dan *total_harga*. Desain database dapat dilihat pada Gambar 4.14 berikut.



Gambar 4.14 Desain database

Masing-masing entitas yaitu user, profil_individu, rekam_kebutuhan, aktifitas dan makanan memiliki atribut yaitu :

Tabel 4.14 Atribut user

Nama Atribut	Tipe	Keterangan
<i>username</i>	Varchar	<i>Primary key</i>
<i>password</i>	Varchar	Menyimpan password user

Tabel 4.15 Atribut profil_individu

Nama Atribut	Tipe	Keterangan
username	Varchar	Primary key
Id_aktifitas	Integer	Primary key
Nama	Varchar	Menyimpan data nama individu
Usia	Integer	Menyimpan data usia individu
Jenis_kelamin	Char	Menyimpan data jenis kelamin individu
Berat_badan	Float	Menyimpan data berat badan individu
Tinggi_badan	Float	Menyimpan data tinggi badan individu

Tabel 4.16 Atribut aktifitas

Nama Atribut	Tipe	Keterangan
Id_aktifitas	Integer	Primary key
aktifitas	Varchar	Mengambil nama aktifitas
rumus	Varchar	Mengambil rumus aktifitas
deskripsi	Varchar	Mengambil deskripsi aktifitas

Tabel 4.17 Atribut makanan

Nama Atribut	Tipe	Keterangan
Id_makanan	Integer	Primary key
Nama_makanan	Varchar	Mengambil nama makanan
Protein	Float	Mengambil nilai protein
Lemak	Float	Mengambil nilai lemak
Karbohidrat	Float	Mengambil nilai karbohidrat
Harga	Float	Mengambil harga makanan

Tabel 4.18 Atribut rekam_kebutuhan

Nama Atribut	Tipe	Keterangan
Id_rk	Integer	Primary key

Tanggal	Date	Primary key
Username	Date	Mengambil data username
Takaran_konsumsi	Varchar	Menyimpan data takaran konsumsi individu pada tanggal tertentu
Total_harga	int	Menyimpan data total harga konsumsi individu pada tanggal tertentu

4.4 Perancangan antarmuka

Rancangan antarmuka dari perangkat lunak untuk permasalahan Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Orang Dewasa Menggunakan Metode Simpleks Big-M dapat dilihat pada Gambar 4.15 hingga 4.17 berikut :

SIGN UP

Username **1.**

Password **2.**

Nama Lengkap **3.**

Usia **4.**

5. JENIS KELAMIN : ☐ Laki-laki ☐ Perempuan

Berat Badan (kg) **6.**

Tinggi Badan (cm) **7.**

Sedentary **8.**

9. DAFTAR !

Deskripsi Aktivitas

10.

Sedentary	Untuk aktivitas ringan atau tidak beraktivitas.
Mild activity level	latihan intensive selama kurang lebih 20 menit, 1 sampai 3 kali per minggu. Contoh aktivitas ini adalah bersepeda, jogging, basket, berenang, dan lain-lain jika tidak melakukan latihan rutin, tetapi memiliki kehidupan yang sibuk yang membutuhkan anda sering berjalan untuk periode waktu yang lama, Anda harus memilih jenis level ini.
Moderate activity level	latihan intensive selama kurang lebih 30 sampai 60 menit, 3 sampai 4 kali per minggu. Contoh aktivitas level ini sama seperti aktivitas pada Mild Activity level.
Heavy or (Labor-intensive) act	latihan intensive selama 60 menit atau lebih, 5 sampai 7 kali per minggu (untuk contoh aktivitas seperti level Mild Activity), pekerjaan berat secara intensive juga memenuhi syarat untuk level ini. Pekerjaan berat ini meliputi pekerjaan konstruksi (tukang batu, tukang kayu, tukang pada umumnya, dan lain-lain.), juga petani atau pekerjaan yang serupa.
Extreme level	aktivitas yang sangat aktif dan aktivitas yang menuntut: contoh yang termasuk aktivitas ini: atlet dengan jadwal latihan yang tak padat dengan banyak sesi latihan sepanjang hari atau beberapa pekerjaan yang menuntut, seperti penambang batubara atau bekerja pada waktu yang lama. Umumnya level aktivitas ini sulit untuk di capai.

Gambar 4.15 Perancangan antarmuka input biodata individu

1. Form input username
2. Form input password
3. Form input nama individu
4. Form input usia
5. Form input jenis kelamin
6. Form input berat badan

7. Form input tinggi badan
8. Form input jenis aktivitas
9. Button menyimpan biodata individu
10. Tabel deskripsi aktivitas

START AKBARKIM
PROFIL
LOG OUT

DATA MENU MAKANAN							
NO	NAMA MAKANAN	KALORI	PROTEIN	LEMAK	KARBOHIDRAT	HARGA	PILIH
1	Nasi Putih	178.00	2.10	0.10	40.60	2300.00	☒
2	Nasi Beras Merah	179.50	3.75	0.45	38.50	2750.00	☒
3	Nasi Beras Hitam	150.00	5.40	0.50	40.00	3500.00	☐
4	Nasi Jagung	149.00	4.10	1.30	30.30	1500.00	☐
5	Papeda	84.60	61.00	0.20	14.90	1750.00	☒
6	Nasi Uduk	232.00	3.00	12.00	28.00	3500.00	☐
7	Nasi Tim daging	250.00	11.00	15.40	56.10	4000.00	☐
8	nasi minyak	265.00	15.00	17.00	62.00	4500.00	☐
9	Nasi Goreng	250.00	6.30	6.23	21.06	3000.00	☐
10	Ketupat	182.00	1.90	0.10	42.00	1000.00	☐
11	Nasi Bakar	195.00	1.80	2.50	48.00	1500.00	☐
12	Nasi Lhvet	365.00	7.13	0.66	79.85	2350.00	☐
13	Buras	88.00	2.30	1.30	16.70	750.00	☐
14	Sereal Beras	432.00	5.43	12.90	73.76	5000.00	☐
15	Nasi Kuning	475.00	7.80	15.80	80.00	3000.00	☐
16	Bubur kacang Hijau	205.00	5.00	3.20	45.00	2000.00	☐

Gambar 4.16 Perancangan antarmuka input pilihan makanan

1. Checkbox makanan yang ingin dipilih

MATRIKS AKHIR (Iterasi ke- 7)

Kolom	Variabel	x1	x2	x3	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6	S9	A7
Cj	Cj	2300.00	2750.00	1750.00	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	10000	0	10000
0	S9	-0.031	0	0	0.017	0	-0.002	0	0	0	0	0	0	0	0	0	0	0	1	-1
0	S2	-0.374	0	0	-0	1	-0.012	0	0	0	0	0	0	0	0	0	0	0	0	0
0	S6	-0	0	0	0	0	1	0	0	0	0	1	-1	0	0	0	0	0	0	0
1750.00	x3	-0.031	0	1	0.017	0	-0.002	0	0	0	0	0	0	0	0	0	0	0	0	0
10000	A2	-0.374	0	0	-0	0	-0.012	0	0	-1	1	0	0	0	0	0	0	0	0	0
0	S8	1.067	0	0	-0.006	0	0.027	0	0	0	0	0	0	0	0	0	1	-1	0	0
10000	A4	-10.813	0	0	-0.257	0	-4.637	0	0	0	0	0	0	1	0	0	0	0	0	0
10000	A5	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0
2750.00	x2	1.067	1	0	-0.006	0	0.027	0	0	0	0	0	0	0	0	0	0	0	0	0
0	S4	0	0	0	1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0

Gambar 4.17 Perancangan antarmuka output matriks makanan

4.5 Perancangan ujicoba dan evaluasi

Untuk dapat mengevaluasi kelayakan program maka diperlukan uji coba untuk mendapatkan nilai parameter yang optimal. Adapun uji coba yang dilakukan antara lain :

1. Pengujian validitas hasil perhitungan.
2. Pengujian sensitivitas nilai minimum konsumsi makanan.
3. Pengujian sensitivitas jenis dan jumlah pilihan makanan

4.5.1 Pengujian validitas hasil perhitungan

Pengujian validitas hasil perhitungan adalah pengujian yang dilakukan untuk mengkoreksi kevalidan hasil output program dengan beberapa fungsi kendala dan fungsi tujuan pada perhitungan manual. Pengujian ini dilakukan dengan cara memasukkan hasil output satuan konsumsi pada tiap makanan ke dalam fungsi kendala , kemudian apakah hasilnya sesuai dengan syarat batas (valid) atau tidak.

4.5.2 Pengujian sensitivitas nilai minimum konsumsi makanan

Pengujian sensitivitas nilai minimum konsumsi makanan adalah pengujian yang dilakukan dengan cara mengubah-ubah nilai batas minimum konsumsi pada setiap makanan. Dengan pengujian ini diharapkan dapat menentukan nilai batas konsumsi yang kemudian menghasilkan output biaya yang paling minimum.

Tabel 4.19 Rancangan pengujian sensitivitas nilai minimum konsumsi makanan

Batas	Makanan			Harga	Total Biaya
	X1	X2	X3		
1				X1 = ;	
0,9				X2 = ;	
0,8				X3 = ;	
0,7					
0,6					
0,5					

4.5.3 Pengujian sensitivitas jenis dan jumlah pilihan makanan

Pengujian sensitivitas jumlah pilihan makanan adalah pengujian yang dilakukan untuk mengetahui pengaruh jumlah pilihan makanan terhadap keluaran. Pengujian ini dilakukan dengan cara mengubah jumlah input makanan yang dikategorikan dalam tiga kategori yaitu makanan yang dipilih secara random, makanan dipilih berdasarkan harga terendah dan makanan dipilih berdasarkan harga tertinggi. Masing-masing akan diuji dalam 6 varian jumlah makanan yaitu 1,2,3,5,7 dan 9. Dengan pengujian ini diharapkan dapat mengetahui pengaruh jumlah pilihan makanan terhadap keluaran yang optimal.

Tabel 4.20 Rancangan pengujian sensitivitas jumlah pilihan makanan

jumlah makanan	id makanan	validitas					Optimal itas
		kalori	protein	lemak	karbohidrat	harga	
1	57	v	x	x	v	x	x
2	57,41						
3	57,41,59						
5	57,41,59,94,58						
7	57,41,59,94,58,55,81						
9	57,41,59,94,58,55,81,52,87						

Keterangan :

X : tidak memenuhi

V : memenuhi

BAB 5 IMPLEMENTASI

5.1 Lingkungan implementasi

Adapun lingkungan implementasinya yaitu lingkungan perangkat keras (hardware) dan perangkat lunak (software). Selanjutnya implementasi program dan implementasi interface.

5.1.1 Lingkungan perangkat keras

Spesifikasi perangkat keras yang digunakan selama proses implementasi metode Simpleks Big-M untuk mengatasi permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa adalah sebagai berikut :

1. Intel core i5-3230M 2,6 GHz
2. RAM 4 GB
3. Hard Disk 500 GB
4. Graphics Intel HD Graphics 4000

5.1.2 Lingkungan perangkat lunak

Spesifikasi perangkat lunak yang digunakan selama proses implementasi metode Simpleks Big-M untuk mengatasi permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa adalah sebagai berikut :

1. OS Windows 10 64-bit sebagai operating system yang digunakan untuk menjalankan semua aplikasi yang dibutuhkan.
2. Microsoft office 2013 sebagai aplikasi penunjang dokumen dan perhitungan.
3. Microsoft visio 2010 sebagai aplikasi penunjang untuk membuat algoritma flowchart.
4. Sublime text 3 sebagai text editor untuk membuat source code.
5. MySQL database sebagai database engine yang dipilih guna penyimpanan data dan pencarian query.
6. XAMPP sebagai aplikasi yang menyediakan service network dan web service pada umumnya.
7. Framework CodeIgniter sebagai framework yang digunakan untuk membangun sistem dalam aplikasi.
8. Google chrome sebagai aplikasi web yang diperlukan untuk runing program.

5.2 Implementasi program

Berdasarkan perancangan perangkat lunak pada Bab IV ,maka pada subbab ini merupakan implementasi program dalam pembuatan aplikasi dengan menggunakan bahasa pemrograman Web Based (PHP, HTML, CSS) dan MySQL. Adapun dalam pengembangan aplikasi digunakan framework Codeigniter maka konsep MVC digunakan. *Class Model* diberi nama *model_user* sedangkan *Class controller* diberi nama *user*. Berikut merupakan implementasi program dari

proses-proses pembuatan aplikasi untuk optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa dengan menggunakan metode Simpleks Big-M .

5.2.1 Implementasi proses input data individu

Langkah awal untuk mengoptimasi pemenuhan biaya dan gizi dengan menggunakan metode Simpleks ini diawali dengan memasukkan data individu ke dalam database yang kemudian akan diolah oleh sistem. Proses implementasi input data individu dapat dilihat pada *sourcecode* 5.1 dan 5.2.

```

1  function aktifitas() {
2      $query=$this->db->query("SELECT * FROM aktifitas");
3      return $query->result() ;
4  }
5
6  //insert user account
7  function insertUser() {
8      $sql = "INSERT INTO user(username,password) VALUES (?,?)";
9      $data = array($this->input->post('username'),
10                  $this->input->post('password'));
11      $query = $this->db->query($sql,$data);
12  }
13
14  //insert profil individu
15  function insertPI() {
16      $sql="INSERT INTO profil_individu (username, nama,
17      usia,jenis_kelamin, berat_badan,tinggi_badan,id_aktifitas)
18      VALUES (?,,?,,?,,?,,?)";
19      $data= array($this->input->post('username'),
20                  $this->input->post('nama'),
21                  $this->input->post('usia'),
22                  $this->input->post('jk'),
23                  $this->input->post('berat'),
24                  $this->input->post('tinggi'),
25                  $this->input->post('aktifitas'));
26      $query = $this->db->query($sql,$data);
27  }

```

Soucecode 5.1 Model input data individu

```

1  public function getsignup() {
2
3      $username    = $this->input->post('username');
4      $password    = $this->input->post('password');
5      $nama        = $this->input->post('nama');

```

```

6      $usia      = $this->input->post('usia');
7      $jeniskelamin = $this->input->post('jk');
8      if ($jeniskelamin == 'Laki-laki') {
9          $jeniskelamin= str_replace($jeniskelamin,'L',
10              $jeniskelamin);
11      }
12      else
13          $jeniskelamin= str_replace($jeniskelamin,'P',
14              $jeniskelamin);
15
16      $beratbadan    = $this->input->post('berat');
17      $tinggibadan   = $this->input->post('tinggi');
18      $aktifitas     = $this->input->post('aktifitas');
19      $check = $this->model_user->checkUser();
20      if($check == "false"){
21          $insertUser = $this->model_user->insertUser();
22          $insertPI   = $this->model_user->insertPI();
23          redirect('user/index','refresh');
24      }
25      else{
26          echo "Akun $username sudah terdaftar";
27      }
28  }
29  }

```

Sourcecode 5.2 Controller input data individu

5.2.2 Implementasi proses pilih makanan

Proses selanjutnya setelah menginputkan data individu adalah memilih makanan yang akan dikonsumsi. Proses implementasi pilih makanan dapat dilihat pada *sourcecode* 5.3 dan 5.4.

```

1  function makanan(){
2      $query=$this->db->query("SELECT * FROM makanan");
3      return $query->result() ;
4  }

```

Sourcecode 5.3 Model pilih makanan

```

1  public function pilihmakanan(){
2
3      $data['username']=$this->session->userdata('username');
4      $data['status']=$this->session->userdata('status');
5      if ($data['status']== 'true') {
6          $data['makanan']= $this->model_user->makanan();
7          $this->load->view('header');

```

```

8      $this->load->view('pilih makanan', $data);
9      $this->load->view('footer');
10     }
11     else{
12         $this->load->view('belumlogin');
13     }
14 }

```

Sourcecode 5.4 Controller pilih makanan

5.2.3 Implementasi proses dapat makanan

Proses dapat makanan adalah proses lanjutan dari proses subbab 5.2.2, dimana setelah menampilkan daftar makanan maka daftar pilihan makanan dipilih akan diambil dari database untuk diolah ke proses-proses berikutnya. Proses implementasi dapat makanan dapat dilihat pada *sourcecode* 5.5 dan 5.6.

```

1  function getMakanan($pilih makanan){
2      $query = $this->db->query("SELECT * FROM makanan
3          WHERE id_makanan = '$pilih makanan' ");
4          return $query->row_array();
5  }

```

Sourcecode 5.5 Model dapat makanan

```

1  private function getMakanan($data,$pilih makanan){
2      foreach ($pilih makanan as $row) {
3          $getMakanan[] = $this->model_user->getMakanan($row);}
4      return $getMakanan ;
5  }
6
7  private function countMakanan($getMakanan){
8      $jumlahMakanan= count($getMakanan);
9      return $jumlahMakanan ;
10 }
11
12 private function countNutrisi($nutrisiuser){
13     $batasNutrisi = count($nutrisiuser);
14     return $batasNutrisi ;
15 }

```

Sourcecode 5.6 Contoller dapat makanan

5.2.4 Implementasi proses menghitung nutrisi individu

Pada proses ini langkah setelah memasukkan inputan, kemudian langkah selanjutnya adalah menghitung kebutuhan nutrisi individu. Adapun proses implementasi menghitung nutrisi individu dapat dilihat pada *sourcecode* 5.7.

```

1 private function hitungNutrisi($data,$pilihMakanan){
2     $profil= $this->model_user->profilku($data['username']);
3     if ($profil[0]->jenis_kelamin == 'L') {
4         $BMR = 88.362 + ( 13.397 * $profil[0]->
5         berat_badan) + ( 4.799 * $profil[0]->tinggi_badan ) -
6         (5.677 * $profil[0]->usia );}
7     else{
8         $BMR = 447.593 + ( 9.247 * $profil[0]->
9         berat_badan) + ( 3.098 * $profil[0]->tinggi_badan ) -
10        (4.330 * $profil[0]->usia );
11    }
12
13    if ($profil[0]->id_aktifitas == 1) {
14        $TEE= $BMR * 1.2 ;
15    }
16    elseif ($profil[0]->id_aktifitas == 2) {
17        $TEE= $BMR * 1.3 -1.375 ;
18    }
19    elseif ($profil[0]->id_aktifitas == 3) {
20        $TEE= $BMR * 1.5 -1.55 ;
21    }
22    elseif ($profil[0]->id_aktifitas == 4) {
23        $TEE= $BMR * 1.7 ;
24    }
25    else{
26        $TEE= $BMR * 1.9 ;
27    }
28
29        $protein_BB = ($TEE * 0.1)/4 ;
30        $protein_BA = ($TEE * 0.35)/4 ;
31        $Carbohidrate_BB = ($TEE * 0.45)/4 ;
32        $Carbohidrate_BA = ($TEE * 0.65)/4 ;
33        $Fat_BB = ($TEE * 0.2)/9 ;
34        $Fat_BA = ($TEE * 0.35)/9 ;
35
36        $nutrisiuser = array('protein_BA' => $protein_BA,
37        'Fat_BA' => $Fat_BA, 'Carbohidrate_BA' =>$Carbohidrate_BA,
38        'protein_BB' => $protein_BB, 'Fat_BB' => $Fat_BB,
39        'Carbohidrate_BB' => $Carbohidrate_BB, 'TEE' => $TEE);
40
41    return $nutrisiuser;

```

42	}
----	---

Soucecode 5.7 Controller menghitung nutrisi individu

5.2.5 Implementasi proses fungsi tujuan

Langkah selanjutnya adalah dengan mengimplementasikan proses fungsi tujuan. Dengan menggunakan inputan yang diperoleh, kemudian inputan makanan kolom harga dimodelkan ke dalam bentuk fungsi. Proses implementasi fungsi tujuan dapat dilihat pada *sourcecode* 5.8.

1	private function fungsiTujuan(\$getMakanan,\$jumlahMakanan){
2	
3	\$Z="";
4	for (\$i=0; \$i < \$jumlahMakanan; \$i++) {
5	if (\$i<\$jumlahMakanan-1) {
6	\$Z=\$Z.\$getMakanan[\$i]['harga']."x".(\$i+1)."+";
7	}
8	} else{
9	\$Z = \$Z.\$getMakanan[\$i]['harga']."x".(\$i+1) ;
10	}
11	}
12	return \$Z ;
13	}

Soucecode 5.8 Controller fungsi tujuan

5.2.6 Implementasi proses fungsi kendala

Langkah selanjutnya adalah dengan mengimplementasikan proses fungsi kendala. Dengan menggunakan inputan yang diperoleh, kemudian inputan makanan kolom kalori, protein, lemak dan karbohidrat dimodelkan ke dalam bentuk fungsi dimana kebutuhan user menjadi batasannya. Adapun implementasi fungsi kendala dapat dilihat pada *sourcecode* 5.9.

1	private function fungsiKendala(\$nutrisiuser,\$jumlahMakanan,
2	\$batasNutrisi, \$getMakanan, \$minKendala){
3	
4	\$batas[]="";
5	for (\$i=0; \$i < \$batasNutrisi; \$i++) {
6	for (\$j=0; \$j < \$jumlahMakanan; \$j++) {
7	if (\$i==0) {
8	if (\$j<\$jumlahMakanan-1) {
9	\$batas[\$i] =
10	\$batas[\$i].\$getMakanan[\$j]['protein']."x".(\$j+1)."+ " ;}
11	} else{
12	\$batas[\$i] =

```

13 $batas[$i].$getMakanan[$j]['protein']."x".($j+1)." <=
14 ".$nutrisiuser['protein_BA'] ;
15     }
16 }
17 else if ($i==1) {
18     if ($j<$jumlahMakanan-1) {
19         @$batas[$i] =
20 $batas[$i].$getMakanan[$j]['lemak']."x".($j+1)."+";}
21     else{
22 $batas[$i] =
23 $batas[$i].$getMakanan[$j]['lemak']."x".($j+1)." <=
24 ".$nutrisiuser['Fat_BA'] ;
25     }
26 }
27
28     else if ($i==2) {
29         if ($j<$jumlahMakanan-1) {
30 @$batas[$i] =
31 $batas[$i].$getMakanan[$j]['karbohidrat']."x".($j+1)."+" ;}
32     else{
33 $batas[$i] =
34 $batas[$i].$getMakanan[$j]['karbohidrat']."x".($j+1)." <=
35 ".$nutrisiuser['Carbohidrate_BA'];
36     }
37 }
38     else if ($i==3) {
39         if ($j<$jumlahMakanan-1) {
40 @$batas[$i] =
41 $batas[$i].$getMakanan[$j]['protein']."x".($j+1)."+" ;}
42     else{
43 $batas[$i] =
44 $batas[$i].$getMakanan[$j]['protein']."x".($j+1)." >=
45 ".$nutrisiuser['protein_BB'] ;
46     }
47 }
48     else if ($i==4) {
49         if ($j<$jumlahMakanan-1) {
50 @$batas[$i] =
51 $batas[$i].$getMakanan[$j]['lemak']."x".($j+1)."+" ; }
52     else{
53 $batas[$i] =

```

```

54 $batas[$i].$getMakanan[$j]['lemak']."x".($j+1)." >=
55 ".$nutrisiuser['Fat_BB'] ;
56     }
57 }
58 else if ($i==5) {
59     if ($j<$jumlahMakanan-1) {
60 @ $batas[$i] =
61 $batas[$i].$getMakanan[$j]['karbohidrat']."x".($j+1)."+ " ;}
62     else{
63 $batas[$i] =
64 $batas[$i].$getMakanan[$j]['karbohidrat']."x".($j+1)." >=
65 ".$nutrisiuser['Carbohidrate_BB'] ;
66     }
67 }
68 else{
69     if ($j<$jumlahMakanan-1) {
70 @ $batas[$i] =
71 $batas[$i].$getMakanan[$j]['kalori']."x".($j+1)."+ " ;}
72     else{
73 $batas[$i] =
74 $batas[$i].$getMakanan[$j]['kalori']."x".($j+1)." =
75 ".$nutrisiuser['TEE'] ;
76     }
77 }
78     } //end for $j
79 } //end for $i
80
81     for ($k=0; $k < $jumlahMakanan ; $k++) {
82 $batas[$k+7] = "x".($k+1)." >=".$minKendala ;
83     }
84     return $batas ;
85 }

```

Soucecode 5.9 Controller fungsi kendala

5.2.7 Implementasi proses model standar

Setelah membuat fungsi tujuan dan fungsi kendala langkah berikutnya adalah memodelkan ke dalam model standar. Model standar ini merupakan penambahan variabel slack, variabel surplus dan variabel *artificial*. Proses implementasi model standar dapat dilihat pada *sourcecode* 5.10.

```

1 private function modelStandar($jumlahMakanan,$batasNutrisi,
2 $batas, $Z, $minKendala){

```

```

3
4     for ($i=0; $i < $jumlahMakanan+4; $i++) {
5         $Z = $Z."+ MA".($i+1) ;
6     }
7
8     for ($j=0; $j < $batasNutrisi+$jumlahMakanan; $j++) {
9         $replace = array(">=", "<=");
10        $s=$j;
11        $a=$j-2 ;
12        if ($j<3) {
13            $batasbaru[$j]=(str_replace($replace,"+ S$s =",
14            $batas[$j]));
15        }
16        else if ($j==6) {
17            $batasbaru[$j]= (str_replace("=", "+ A$a =", $batas[$j]));
18        }
19        else{
20            $batasbaru[$j]= (str_replace($replace, "- S$s + A$a =",
21            $batas[$j]));
22        }
23    }
24    return array($Z,$batasbaru) ;
25 }

```

Soucecode 5.10 Controller model standar

5.2.8 Implementasi proses membuat tabel Simpleks Big-M awal

Langkah awal dari perhitungan Simpleks Big-M adalah mengubah dari model standar ke dalam bentuk tabel awal matriks. Adapun proses implementasi membuat Tabel Simpleks Big-M awal dapat dilihat pada *sourcecode* 5.11.

```

1     private function TabelMatriksAwal($nutrisiuser,
2     $jumlahMakanan, $getMakanan, $minKendala){
3
4         $M = 10000 ;
5         $Jum_KolomSurplus = 10;
6         $Jum_KolomSurplus_flex = 2* $jumlahMakanan;
7         $Jum_Kolom = $jumlahMakanan + $Jum_KolomSurplus +
8         $Jum_KolomSurplus_flex + 1 ;
9         $Jum_Baris = 7 + $jumlahMakanan ;
10
11        $mIdentity = array();
12        //membuat matriks identitas

```

```

13     for ($i=0; $i < $jumlahMakanan; $i++) {
14         for ($j=0; $j < $jumlahMakanan; $j++) {
15             if ($i == $j) {
16                 $mIdentity[$i][$j] = 1;
17             } else {
18                 $mIdentity[$i][$j] = 0;
19             }
20         }
21     }
22
23
24     //mengisi matriks kolom flesibel 1
25     for ($i=-1; $i <= $Jum_Baris; $i++) {
26         for ($j=-1; $j <= $jumlahMakanan; $j++) {
27             if ($i== -1 && $j== -1) {
28                 $MATRIKS[$i][$j] = "Kolom" ;
29             }
30             else if ($i== -1 && $j==0) {
31                 $MATRIKS[$i][$j] = "Variabel" ;
32             }
33             else if (($i== -1) and (0<$j and $j<=$jumlahMakanan)){
34                 $MATRIKS[$i][$j] = "x$j" ;
35             }
36             else if ($i==0 && $j<1) {
37                 $MATRIKS[$i][$j] = "Cj" ;
38             }
39             else if (($i==0) && (0<$j and $j<=$jumlahMakanan)) {
40                 $MATRIKS[$i][$j] = $getMakanan[$j-1]['harga'] ;
41             }
42             else if ((0<$i and $i<4) && ($j== -1)) {
43                 $MATRIKS[$i][$j] = 0 ;
44             }
45             else if ((3<$i and $i<=$Jum_Baris) && ($j== -1)) {
46                 $MATRIKS[$i][$j] = $M ;
47             }
48             else if ((0<$i and $i<4) && ($j==0)) {
49                 $MATRIKS[$i][$j] = "S$i" ;
50             }
51             else if ((3<$i and $i<=$Jum_Baris) && ($j==0)) {
52                 $a = $i -3;
53                 $MATRIKS[$i][$j] = "A$a" ;

```

```

54     }
55     else if(($i==1 || $i== 4) && (0<$j and
56         $j<=$jumlahMakanan)){
57         $MATRIKS[$i][$j] = $getMakanan[$j-1]['protein'] ;
58     }
59     else if(($i==2 || $i== 5) && (0<$j and
60         $j<=$jumlahMakanan)){
61         $MATRIKS[$i][$j] = $getMakanan[$j-1]['lemak'] ;
62     }
63     else if(($i==3 || $i== 6) && (0<$j and
64         $j<=$jumlahMakanan)){
65         $MATRIKS[$i][$j] = $getMakanan[$j-1]['karbohidrat'] ;
66     }
67     else if(($i==7) && (0<$j and $j<=$jumlahMakanan)){
68         $MATRIKS[$i][$j] = $getMakanan[$j-1]['kalori'] ;
69     }
70     else {
71         $MATRIKS[$i][$j] = 0 ;
72     }
73 }
74 }
75
76 // mengisi matriks identitas pada baris ke-8 dst
77 $mi = 0;
78 for ($i=8; $i <=$Jum_Baris; $i++) {
79     $mj = 0 ;
80     for ($j=1; $j <= $jumlahMakanan; $j++) {
81         $MATRIKS[$i][$j] = $mIdentity[$mi][$mj];
82         $mj++;
83     }$mi++;
84 }
85
86 //mengisi matriks kolom 2 statis ()
87 $var_basis = array("S1", "S2", "S3", "S4", "A1", "S5",
88     "A2", "S6","A3","A4");
89 $scj_basis = array(0,0,0,0,$M,0,$M,0,$M,$M) ;
90 $nilai_basis = array(array(1,0,0,0,0,0,0,0,0,0),
91     array(0,1,0,0,0,0,0,0,0,0),
92     array(0,0,1,0,0,0,0,0,0,0),
93     array(0,0,0,-1,1,0,0,0,0,0),
94     array(0,0,0,0,0,-1,1,0,0,0),

```

```

95         array(0,0,0,0,0,0,0,-1,1,0),
96         array(0,0,0,0,0,0,0,0,0,1));
97     $bataskebutuhan = array($nutrisiuser['protein_BA'],
98     $nutrisiuser['Fat_BA'], $nutrisiuser['Carbohidrate_BA'],
99     $nutrisiuser['protein_BB'], $nutrisiuser['Fat_BB'],
100    $nutrisiuser['Carbohidrate_BB'], $nutrisiuser['TEE']);
101
102    for ($i=-1; $i <= $Jum_Baris; $i++) {
103        for ($j=$jumlahMakanan+1; $j <= $jumlahMakanan+10;
104        $j++) {
105            if (($i== -1) and ($jumlahMakanan<$j and
106            $j<=$jumlahMakanan+10)) {
107                $MATRIKS[$i][$j] = $var_basis[$j-$jumlahMakanan-1];
108            }
109            else if (($i==0) and ($jumlahMakanan<$j and
110            $j<=$jumlahMakanan+10)) {
111                $MATRIKS[$i][$j] = $cj_basis[$j-$jumlahMakanan-1];
112            }
113            else if ((0<$i and $i<=7) and ($jumlahMakanan<$j and
114            $j<=$jumlahMakanan+10)) {
115                $MATRIKS[$i][$j] = $nilai_basis[$i-1][$j-
116                $jumlahMakanan-1] ;
117            }
118            else{
119                $MATRIKS[$i][$j] = 0 ;
120            }
121        }
122    }
123
124    //mengisi matriks kolom flesibel 3
125    $initS = 7;
126    $initA = 5;
127
128    for ($i=-1; $i <= $Jum_Baris; $i++) {
129        for ($j=$jumlahMakanan+11; $j <= $Jum_Kolom; $j++) {
130            if (($i== -1) and ($jumlahMakanan+10<$j and
131            $j<$Jum_Kolom )) {
132                if (substr($MATRIKS[$i][$j-1], 0, 1) == "A") {
133                    $MATRIKS[$i][$j] = "S".($initS);
134                    $initS++;
135                } else {

```

```

136      $MATRIKS[$i][$j] = "A".($initA);
137      $initA++;
138  }
139  }
140  else if (($i==0) and ($jumlahMakanan+10<$j and
141  $j<$Jum_Kolom)) {
142      if (substr($MATRIKS[-1][$j],0,1 ) == "A") {
143          $MATRIKS[$i][$j] = $M ;
144      }
145      else{
146          $MATRIKS[$i][$j] = 0 ;
147      }
148  }
149  else if ((0<$i and $i<=7) and
150  ($jumlahMakanan+10<$j and $j<$Jum_Kolom)) {
151      $MATRIKS[$i][$j] = 0 ;
152  }
153  else if ($i > 7 and $i<=$Jum_Baris and $j <
154  $Jum_Kolom) {
155      if($MATRIKS[$i][0] === $MATRIKS[-1][$j+1]) {
156          $MATRIKS[$i][$j] = -1;
157          $MATRIKS[$i][$j+1] = 1;
158      } else {
159          $MATRIKS[$i][$j] = 0;
160          $MATRIKS[$i][$j+1] = 0;
161      }
162      $j++;
163  }
164  else if ((0<$i and $i<=7) and
165  ($j==$Jum_Kolom)) {
166      $MATRIKS[$i][$j] = $bataskebutuhan[$i-1] ;
167  }
168  else if ((7<$i and $i<=$Jum_Baris) and
169  ($j==$Jum_Kolom)) {
170      $MATRIKS[$i][$j] = $minKendala;
171  }
172  else{
173      $MATRIKS[$i][$j] = "BATAS" ;
174  }
175  }
176  }

```

```

177     }
178
179     return array($MATRIKS,$Jum_Baris,$Jum_Kolom) ;
180 }

```

Soucecode 5.11 Controller membuat tabel awal

5.2.9 Implementasi proses menghitung Zj dan Cj-Zj

Langkah berikutnya adalah menentukan nilai Zj dan Cj-Zj. Proses implementasi menghitung Zj dan Cj-Zj ditunjukkan pada *sourcecode* 5.12.

```

1  private function cjzj ($MATRIKS,$Jum_Baris,$Jum_Kolom){
2
3      $Zj = array();
4      $CjZj = array();
5
6      for ($i=1; $i <= $Jum_Kolom; $i++) {
7          $Zj[$i] = 0 ;
8          for ($j=1; $j <= $Jum_Baris ; $j++) {
9              $Zj[$i] += ($MATRIKS[$j][-1] * $MATRIKS[$j][$i]);
10             if ($i>0 and $i<$Jum_Kolom) {
11                 $CjZj[$i] = $MATRIKS[0][$i] - $Zj[$i] ;
12             }
13             else{
14                 $CjZj[$i] = null ;
15             }
16         }
17     }
18     return array($Zj,$CjZj);
19 }

```

Soucecode 5.12 Controller menghitung Zj dan Cj-Zj

5.2.10 Implementasi proses menentukan kolom kunci

Setelah didapat nilai Zj dan Cj-Zj maka langkah selanjutnya adalah menentukan kolom kunci. Kolom kunci didapat dari nilai Cj-Zj negatif yang paling besar. Adapun proses implementasi menentukan kolom kunci dapat dilihat pada *sourcecode* 5.13.

```

1  private function kolomPengganti ($MATRIKS, $Jum_Baris,
2      $Jum_Kolom, $Zj,$CjZj){
3
4      $CjZjKolTerpilih = $CjZj[1];
5      $indexKolTerpilih = 1;
6      for ($i=1; $i <$Jum_Kolom ; $i++) {

```

```

7         if ($CjZjKolTerpilih > $CjZj[$i]) {
8             $CjZjKolTerpilih = $CjZj[$i] ;
9             $indexKolTerpilih = $i ;
10        }
11    }
12    return array($CjZjKolTerpilih,$indexKolTerpilih)
13 }

```

Sourcecode 5.13 Controller menentukan kolom kunci

5.2.11 Implementasi proses menentukan baris kunci

Indeks kolom kunci yang didapat pada subbab 5.2.10 digunakan untuk menentukan baris kunci. Baris kunci dipilih berdasarkan nilai bagi antara kolom batas dengan kolom kunci. Kemudian dipilih baris yang memiliki nilai positif terkecil. Proses implementasi menentukan baris kunci dapat dilihat pada *sourcecode 5.14*.

```

1 private function barisTerganti($MATRIKS, $Jum_Baris,
2 $Jum_Kolom, $indexKolTerpilih){
3
4     $hasilBagi = array();
5     $barTerpilih =10000000000000;
6     $indexBarTerpilih = 1;
7     for ($i=1; $i <= $Jum_Baris ; $i++) {
8         $namakolom[$i] = $MATRIKS[$i][0] ;
9         $kolomterpilih[$i] = $MATRIKS[$i][$indexKolTerpilih] ;
10        $batas[$i] = $MATRIKS[$i][$Jum_Kolom] ;
11        if ($kolomterpilih[$i] != 0) {
12            $hasilBagi[$i] = $batas[$i] / $kolomterpilih[$i] ;
13            if ($hasilBagi[$i] > 0) {
14                if ($barTerpilih > $hasilBagi[$i]) {
15                    $barTerpilih = $hasilBagi[$i] ;
16                    $indexBarTerpilih = $i ;
17                }
18            }
19        }
20        else{
21            $hasilBagi[$i] = "infinity" ;
22        }
23    } //end of for
24    Return array($hasilBagi, $barTerpilih,
25        $indexBarTerpilih);
26 }

```

Soucecode 5.14 Controller menentukan baris kunci

5.2.12 Implementasi proses revisi koefisien baru

Langkah setelah menentukan kolom kunci dan baris kunci adalah merevisi koefisien baris terpilih/baris kunci maupun baris yang lain. Adapun proses implementasi revisi koefisien baru dapat dilihat pada *sourcecode* 5.15.

```

1 private function revKoefisienBaru($MATRIKS, $Jum_Baris,
2   $Jum_Kolom, $indexKolTerpilih, $indexBarTerpilih){
3
4     //menghitung koefisien terpilih
5     for ($i=-1; $i <=$Jum_Kolom ; $i++) {
6       if ($i>0) {
7         $MATRIKS_BARU[$indexBarTerpilih][$i] =
8         $MATRIKS[$indexBarTerpilih][$i] /
9         $MATRIKS[$indexBarTerpilih][$indexKolTerpilih] ;
10      }
11      elseif ($i == 0) {
12        $MATRIKS_BARU[$indexBarTerpilih][$i] =
13        $MATRIKS[-1][$indexKolTerpilih] ;
14      }
15      else{
16        $MATRIKS_BARU[$indexBarTerpilih][$i] =
17        $MATRIKS[0][$indexKolTerpilih] ;
18      }
19    }
20    //revisi koefisien baris lainnya
21    for ($i=-1; $i <= $Jum_Baris ; $i++) {
22      for ($j=-1; $j <= $Jum_Kolom ; $j++) {
23        if (($i != $indexBarTerpilih and $i>0) and ($j>0 and
24          $j<=$Jum_Kolom)) {
25          $MATRIKS_BARU[$i][$j] = $MATRIKS[$i][$j] -
26          ($MATRIKS_BARU[$indexBarTerpilih][$j] *
27          $MATRIKS[$i][$indexKolTerpilih] ) ;
28        }
29        else if (($i != $indexBarTerpilih and $i>0 ) and
30          ($j==0 or $j==--1)) {
31          $MATRIKS_BARU[$i][$j] = $MATRIKS[$i][$j] ;
32        }
33        else if ($i==0 or $i==--1) {
34          $MATRIKS_BARU[$i][$j] = $MATRIKS[$i][$j] ;
35        }

```

```

36     else{
37         $MATRIKS_BARU[$i][$j];
38     }
39 }
40 }
41
42 return $MATRIKS_BARU ;
43 }

```

Soucecode 5.15 Controller revisi koefisien baru

5.2.13 Implementasi proses cek positif nilai CjZj

Proses cek positif nilai CjZj adalah proses untuk mengetahui bahwa proses perhitungan dengan metode Simpleks Big-M sudah optimal atau belum. Jika nilai CjZj sudah tidak ada yang bernilai negatif maka proses iterasi sudah optimal dan dihentikan. Proses implementasi cek positif nilai CjZj dapat dilihat pada *sourcecode* 5.16.

```

1 private function cekPositif($array){
2     $jum_positif = 0;
3     for ($i=1; $i < count($array) ; $i++) {
4         if ($array[$i] < 0) {
5             return false;
6         }
7         else {
8             $jum_positif++;
9         }
10    }
11    if ($jum_positif == (count($array) - 1)) {
12        return true;
13    } else {
14        return false;
15    }
16 }

```

Soucecode 5.16 Controller cek positif nilai CjZj

5.2.14 Implementasi proses buat result

Langkah ini digunakan untuk menyimpulkan dan dan menyimpan hasil dari Tabel Simpleks menjadi Tabel hasil sehingga user dapat lebih mudah untuk membaca keluaran. Adapun proses implementasi buat *result* dapat dilihat pada *sourcecode* 5.17.

```

1 private function result($MATRIKS, $Jum_Kolom, $Jum_Baris,
2     $getMakanan, $jumlahMakanan){

```

```

3      $hasil = array() ;
4      $cek = array() ;
5      $x = array() ;
6      $count = 0 ;
7
8      //memasukkan identitas x dan nama
9      for ($i=0; $i < $jumlahMakanan; $i++) {
10         $x[$i][0] = "x".($i+1);
11         $x[$i][1] = $getMakanan[$i]['nama_makanan'] ;
12     }
13
14     //cek apakah ada var x dalam matriks
15     for ($i=1; $i <= $Jum_Baris ; $i++) {
16         if (substr($MATRIKS[$i][0], 0, 1) == "x") {
17             $cek[$count][0] = $MATRIKS[$i][0];
18             $cek[$count][1] = $MATRIKS[$i][$Jum_Kolom];
19             $count++;
20         }
21     }
22
23     $temp = array();
24     for ($i=0; $i < $jumlahMakanan; $i++) {
25         $flag = false;
26         for ($j=0; $j < $count ; $j++) {
27             if ($x[$i][0] == $cek[$j][0]) {
28                 $flag = true;
29                 $temp[0] = $x[$i][0] ;
30                 $temp[1] = $x[$i][1] ;
31                 $temp[2] = $cek[$j][1];
32             }
33         }
34         if ($flag == true){
35             $hasil[$i][0] = $temp[0] ;
36             $hasil[$i][1] = $temp[1] ;
37             $hasil[$i][2] = $temp[2];
38         }
39         else{
40             $hasil[$i][0] = $x[$i][0] ;
41             $hasil[$i][1] = $x[$i][1] ;
42             $hasil[$i][2] = 0;
43         }

```

```

44     }
45     return $hasil ;
46 }

```

Sourcecode 5.17 Controller menampilkan result

5.2.15 Implementasi proses main program

Fungsi main program digunakan untuk memanggil semua fungsi proses perhitungan dengan metode Simpleks Big-M. Dari menghitung kebutuhan individu, memodelkan dalam bentuk pemrograman linear, menghitung dengan metode Simpleks Big-M hingga membuat result. Proses implementasi *main program* dapat dilihat pada *sourcecode* 5.18.

```

1  public function nutrisiuser(){
2      $data['username']=$this->session->userdata('username');
3      $data['status']=$this->session->userdata('status');
4      $data['pilihmaksanan']=$this->input->post("pilihmaksanan");
5      $data['minKendala'] = 1;
6
7      if ($data['status']== 'true') {
8          $data['kebutuhanUser'] = $this->
9              hitungNutrisi($data,$data['pilihmaksanan']);
10         $data['getMaksanan'] = $this->
11             getMaksanan($data,$data['pilihmaksanan']);
12         $data['countMaksanan'] = $this->
13             countMaksanan($data['getMaksanan']) ;
14         $data['countNutrisi'] = $this->
15             countNutrisi($data['kebutuhanUser']) ;
16         $data['fungsiTujuan'] = $this->
17             fungsiTujuan($data['getMaksanan'],$data['countMaksanan']);
18         $data['fungsiKendala'] = $this->
19             fungsiKendala($data['kebutuhanUser'],$data['countMaksanan'],
20                 $data['countNutrisi'],$data['getMaksanan'],$data['minKendala
21                 ']);
22         $modelStandar = $this->
23             modelStandar($data['countMaksanan'],$data['countNutrisi'],
24                 $data['fungsiKendala'],$data['fungsiTujuan'],
25                 $data['minKendala']);
26         $data['modelStandar_Tujuan'] = $modelStandar[0] ;
27         $data['modelStandar_Kendala'] = $modelStandar[1] ;
28
29         // fungsi Tabel matriks awal
30         $TabelMatriksAwal = $this->

```

```

31 TabelMatriksAwal($data['kebutuhanUser'],
32 $data['countMakanan'], $data['getMakanan'],
33 $data['minKendala']);
34     $data['MATRIKS'] = $TabelMatriksAwal[0];
35     $data['MATRIKS_AWAL'] = $TabelMatriksAwal[0];
36     $data['Jum_Baris'] = $TabelMatriksAwal[1];
37     $data['Jum_Kolom'] = $TabelMatriksAwal[2];
38
39     $data['countIterasi'] = 0;
40     do {
41         $zj_cjzj= $this->
42         cjzj($data['MATRIKS'], $data['Jum_Baris'],
43         $data['Jum_Kolom'] );
44         $data['Zj'] = $zj_cjzj[0];
45         $data['CjZj'] = $zj_cjzj[1];
46
47         // minimum negatif CjZj terbesar(kolom pengganti)
48         $kolomPengganti = $this->kolomPengganti($data['MATRIKS'],
49         $data['Jum_Baris'],$data['Jum_Kolom'],$data['Zj'],
50         $data['CjZj']) ;
51         $data['CjZjKolTerpilih'] = $kolomPengganti[0] ;
52         $data['indexKolTerpilih'] = $kolomPengganti[1] ;
53
54         // mencari baris yang terganti
55         $barisTerganti = $this->barisTerganti($data['MATRIKS'],
56         $data['Jum_Baris'],$data['Jum_Kolom'],
57         $data['indexKolTerpilih']) ;
58         $data['hasilBagi'] =$barisTerganti[0] ;
59         $data['barTerpilih'] = $barisTerganti[1] ;
60         $data['indexBarTerpilih'] = $barisTerganti[2] ;
61
62         //koefisien baru yang masuk
63         $data['MATRIKS'] = $this->revKoefisienBaru
64         ($data['MATRIKS'],$data['Jum_Baris'],
65         $data['Jum_Kolom'],$data['indexKolTerpilih'],
66         $data['indexBarTerpilih']);
67         $data['countIterasi']++;
68         $data['CjZj'] = $zj_cjzj[1];
69     }
70     while($this->cekPositif($data['CjZj']) == false);
71     $data['result'] = $this->

```

```

72 result($data['MATRIKS'],$data['Jum_Kolom'],
73 $data['Jum_Baris'],$data['getMakanan'],
74 $data['countMakanan']);
75
76 $this->load->view('header');
77 $this->load->view('nutrisiuser', $data);
78 $this->load->view('footer') ;
79 }
80 else{
81 $this->load->view('belumlogin');
82 }
83 }

```

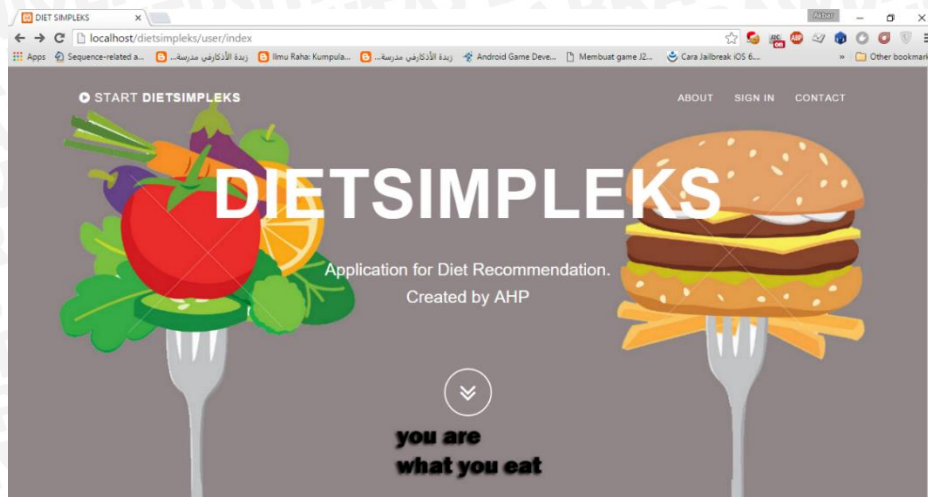
Soucecode 5.18 Contoller main program

5.3 Implementasi antarmuka

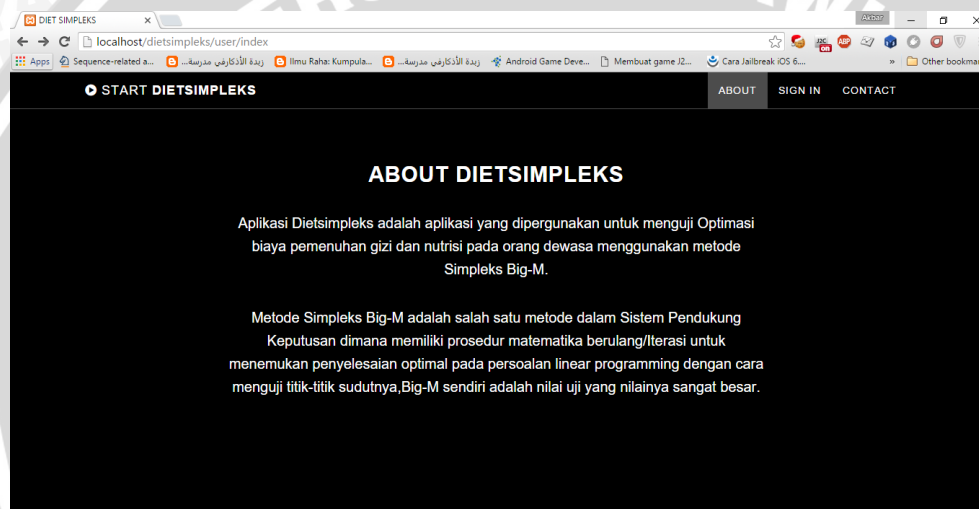
Halaman Antarmuka pada sistem terdiri dari 8 halaman yaitu halaman *index* yang berisi menu *sign in* , *about* dan *contact*. Halaman *Sign up*, halaman *home*, halaman menu makanan, halaman rekam konsumsi, halaman *profil* , halaman pilih makanan dan halaman nutrisi user yang berisi output proses perhitungan.

5.3.1 Halaman index

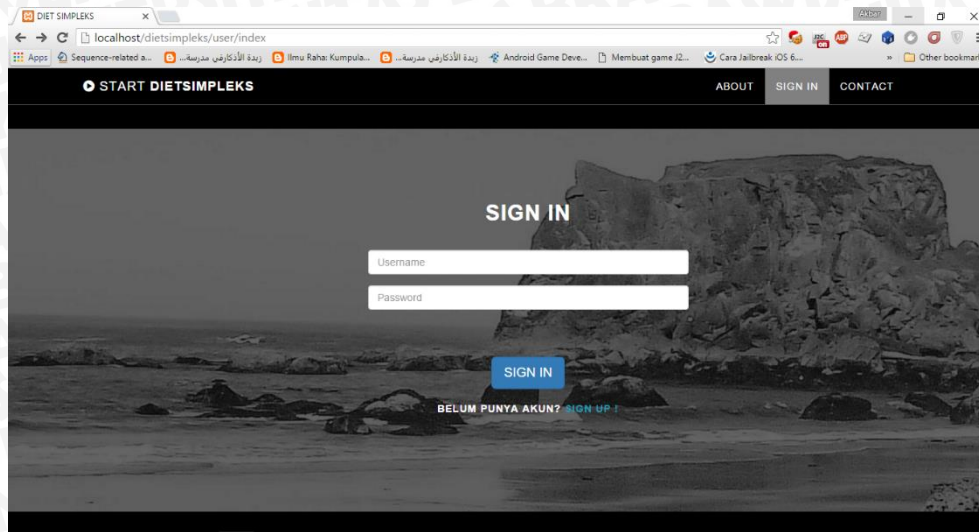
Halaman index adalah halaman awal ketika kita mengakses URL dari aplikasi. Halaman index merupakan halaman awal sebelum user bisa masuk ke dalam sistem. Halaman index terdapat 3 tab menu. Yaitu **About** yang berisi penjelasan mengenai program. **Sign in** yang digunakan user untuk masuk ke dalam sistem. **Contact** yang berisikan contact developer untuk dapat memberikan kritik maupun saran kepada developer. Implementasi halaman index dapat dilihat pada Gambar 5.1 sampai 5.4.



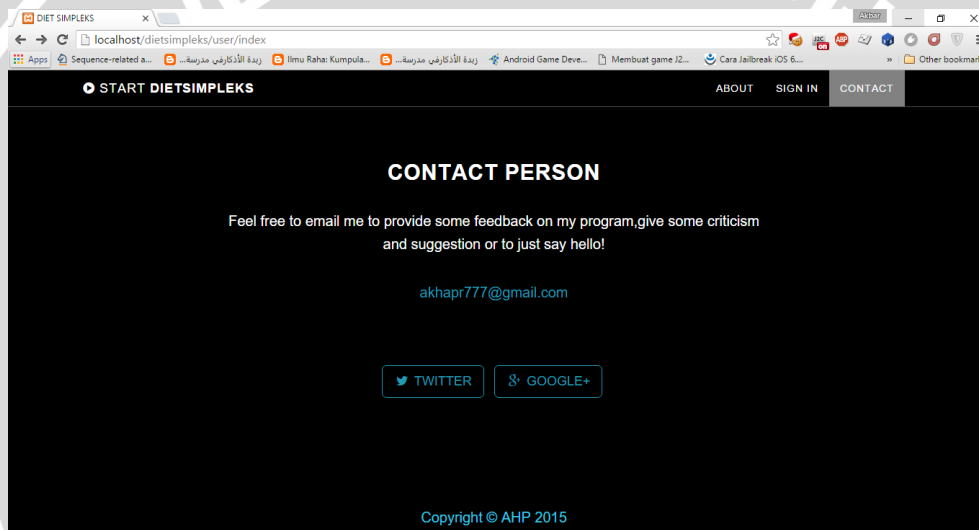
Gambar 5.1 Tampilan halaman *index (menu)*.



Gambar 5.2 Tampilan halaman *index (about)*.



Gambar 5.3 Tampilan halaman *index (sign in)*.



Gambar 5.4 Tampilan halaman *index (contact)*.

5.3.2 Halaman sign up

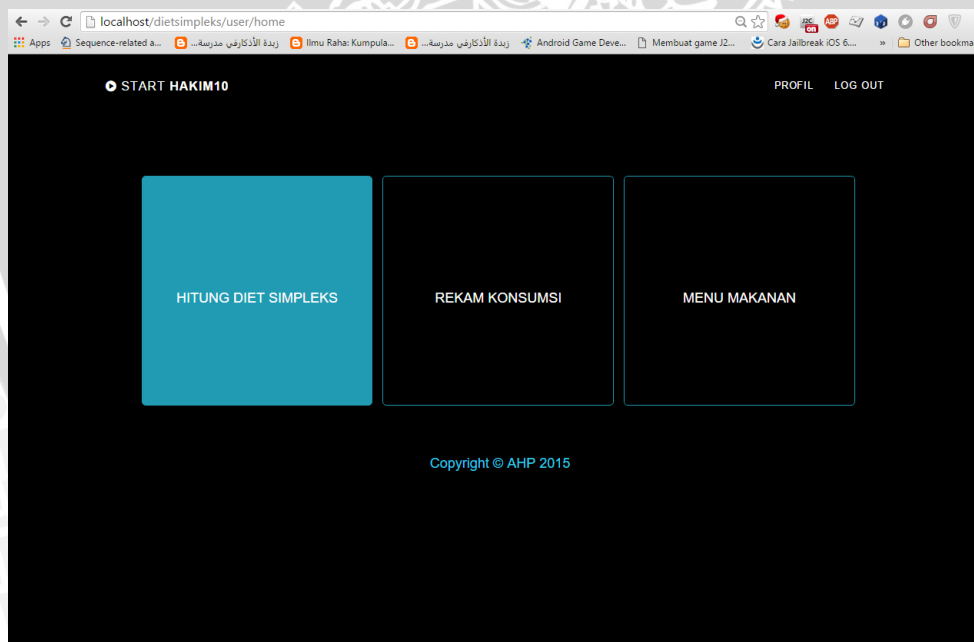
Halaman sign up digunakan untuk menginputkan data pribadi dari individu yang kemudian akan digunakan untuk masuk ke dalam sistem dan dapat melakukan aktivitas lainnya dalam program. Implementasi halaman sign up dapat dilihat pada Gambar 5.5.

Sedentary	Untuk aktivitas ringan atau tidak beraktivitas.
Mild activity level	latihan intensif selama kurang lebih 20 menit, 4 sampai 3 kali per minggu. Contoh aktivitas ini adalah bersepeda, jogging, basket, berenang, dan lain-lain jika tidak melakukan latihan rutin, tetapi memiliki kehidupan yang sibuk yang membutuhkan anda sering berjalan untuk periode waktu yang lama, Anda harus memilih jenis level ini.
Moderate activity level	latihan intensif selama kurang lebih 30 sampai 60 menit, 3 sampai 4 kali per minggu. Contoh aktivitas level ini sama seperti aktivitas pada Mild Activity level.
Heavy or (labor-intensive) act	latihan intensif selama 60 menit atau lebih, 5 sampai 7 kali per minggu (untuk contoh aktivitas seperti level Mild Activity), pekerjaan berat secara intensif juga memenuhi syarat untuk level ini. (Pekerjaan berat ini meliputi pekerjaan konstruksi (tukang betu, tukang kayu, tukang pada umumnya, dan lain-lain). juga petani atau pekerjaan yang serupa.
Extreme level	aktivitas yang sangat aktif dan aktivitas yang menuntut: contoh yang termasuk aktivitas ini: atlit dengan jadwal latihan yang tak padat dengan banyak sesi latihan sepanjang hari atau beberapa pekerjaan yang menuntut, seperti penambang batubara atau bekerja pada waktu yang lama. Umumnya level aktivitas ini sulit untuk di capai.

Gambar 5.5 Tampilan halaman *sign up*.

5.3.3 Halaman home

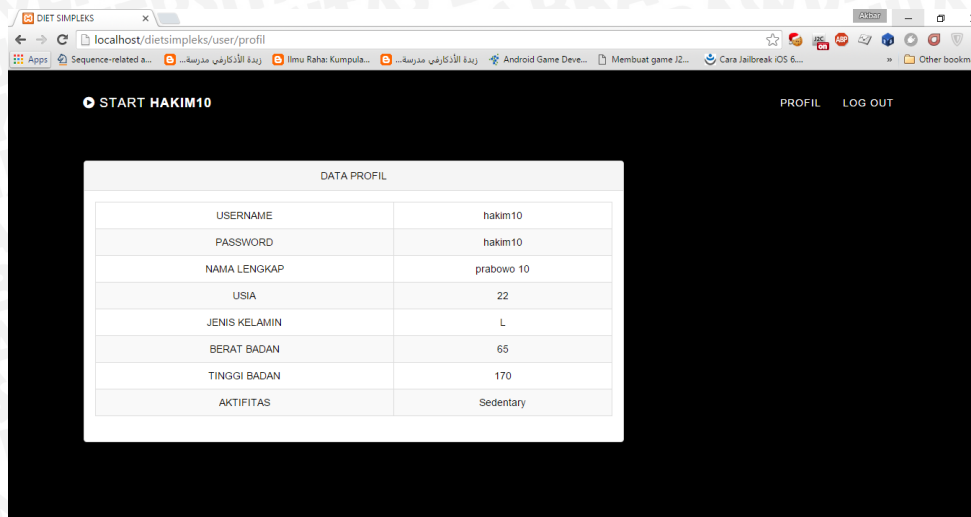
Halaman home berisi beberapa fitur menu yang dapat dipilih. Antara lain Hitung Diet Simpleks, Menu Makanan dan Profil. Implementasi halaman home dapat dilihat pada Gambar 5.6.



Gambar 5.6 Tampilan halaman *home*.

5.3.4 Halaman profil

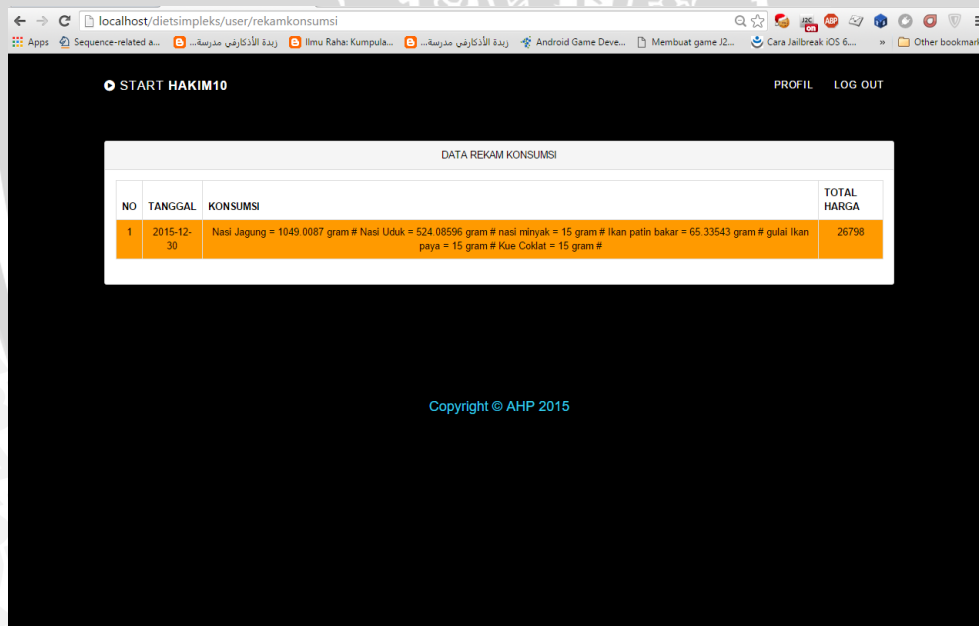
Halaman profil berisikan data dari user yang telah didaftarkan sebelumnya, user hanya memiliki otoritas melihat data profil yang telah dimasukkan. Implementasi halaman profil dapat dilihat pada Gambar 5.7.



Gambar 5.7 Tampilan halaman profil.

5.3.5 Halaman rekam konsumsi

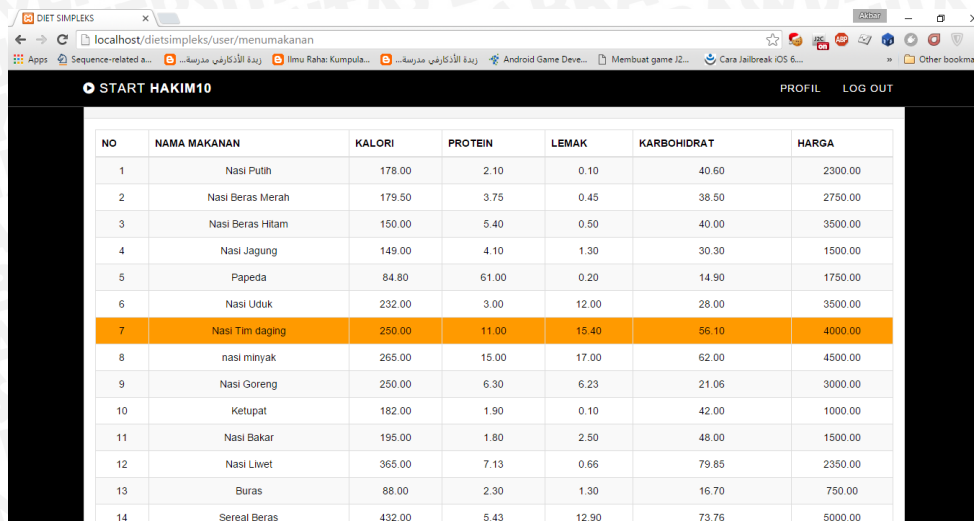
Halaman rekam konsumsi berisikan tampilan makanan yang pernah dikonsumsi oleh user dan disimpan di dalam database. Implementasi halaman rekam konsumsi dapat dilihat pada Gambar 5.8.



Gambar 5.8 Tampilan rekam konsumsi

5.3.6 Halaman menu makanan

Halaman menu makanan berisikan tampilan seluruh menu makanan yang menampilkan nutrisi dari makanan beserta harganya. Pada halaman ini user hanya memiliki otoritas untuk membaca saja. Implementasi halaman menu makanan dapat dilihat pada Gambar 5.9.

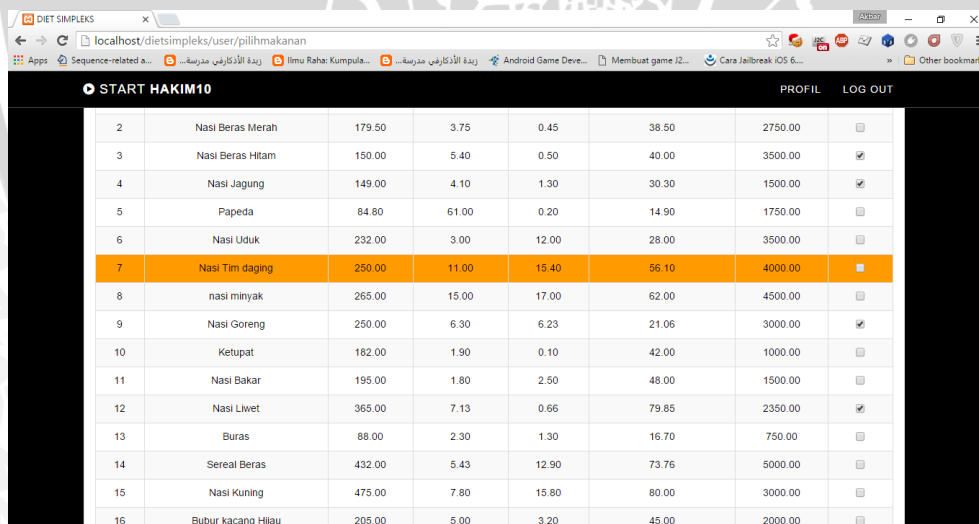


NO	NAMA MAKANAN	KALORI	PROTEIN	LEMAK	KARBOHIDRAT	HARGA
1	Nasi Putih	178.00	2.10	0.10	40.60	2300.00
2	Nasi Beras Merah	179.50	3.75	0.45	38.50	2750.00
3	Nasi Beras Hitam	150.00	5.40	0.50	40.00	3500.00
4	Nasi Jagung	149.00	4.10	1.30	30.30	1500.00
5	Papeda	84.80	61.00	0.20	14.90	1750.00
6	Nasi Uduk	232.00	3.00	12.00	28.00	3500.00
7	Nasi Tim daging	250.00	11.00	15.40	56.10	4000.00
8	nasi minyak	265.00	15.00	17.00	62.00	4500.00
9	Nasi Goreng	250.00	6.30	6.23	21.06	3000.00
10	Ketupat	182.00	1.90	0.10	42.00	1000.00
11	Nasi Bakar	195.00	1.80	2.50	48.00	1500.00
12	Nasi Liwet	365.00	7.13	0.66	79.85	2350.00
13	Buras	88.00	2.30	1.30	16.70	750.00
14	Sereal Beras	432.00	5.43	12.90	73.76	5000.00

Gambar 5.9 Tampilan halaman menu makanan.

5.3.7 Halaman pilih makanan

Halaman pilih makanan merupakan tampilan input bagi user untuk dapat memilih makanan yang ingin dikonsumsi. Halaman ini dapat diakses jika user melakukan klik button menu Hitung Diet Simpleks pada halaman home. Setelah selesai memilih makanan yang diinginkan kemudian user dapat melanjutkan proses hitung dengan melakukan klik pada button SUBMIT yang kemudian akan di direct ke halaman nutrisius. Implementasi halaman pilih makanan dapat dilihat pada Gambar 5.10.



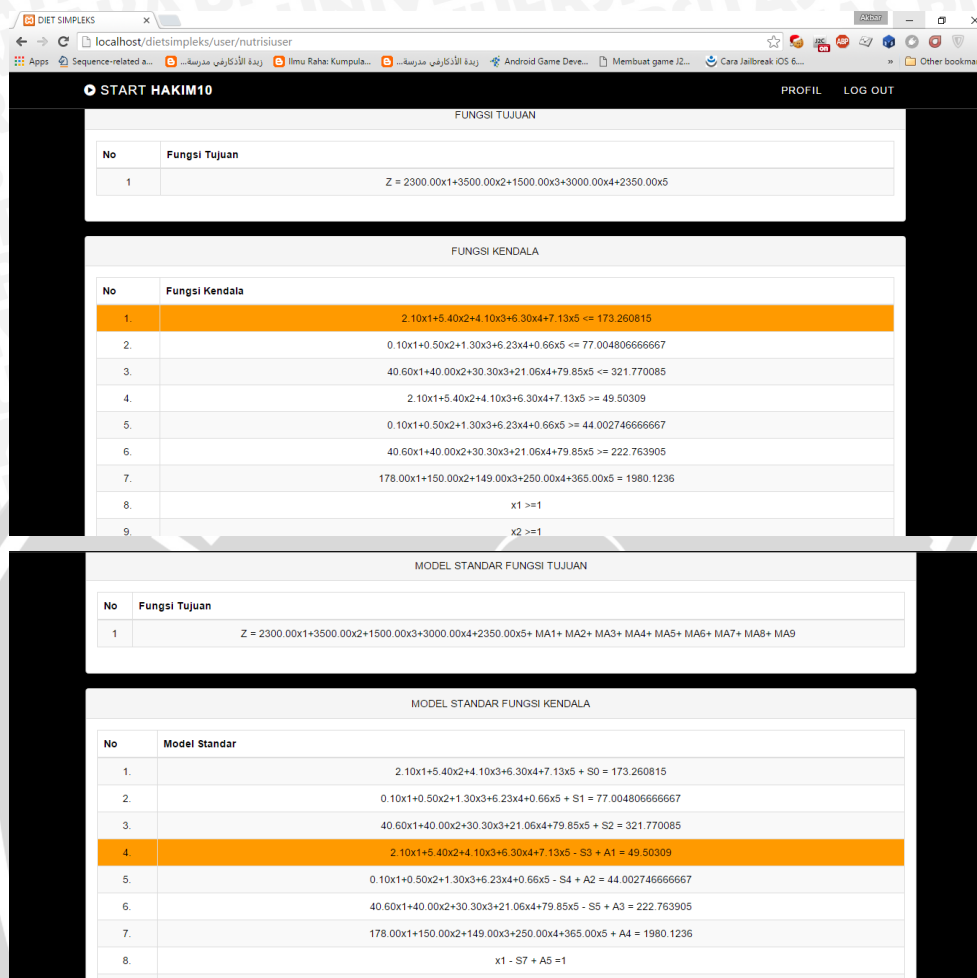
NO	NAMA MAKANAN	KALORI	PROTEIN	LEMAK	KARBOHIDRAT	HARGA	
2	Nasi Beras Merah	179.50	3.75	0.45	38.50	2750.00	☐
3	Nasi Beras Hitam	150.00	5.40	0.50	40.00	3500.00	☒
4	Nasi Jagung	149.00	4.10	1.30	30.30	1500.00	☒
5	Papeda	84.80	61.00	0.20	14.90	1750.00	☐
6	Nasi Uduk	232.00	3.00	12.00	28.00	3500.00	☐
7	Nasi Tim daging	250.00	11.00	15.40	56.10	4000.00	☒
8	nasi minyak	265.00	15.00	17.00	62.00	4500.00	☐
9	Nasi Goreng	250.00	6.30	6.23	21.06	3000.00	☒
10	Ketupat	182.00	1.90	0.10	42.00	1000.00	☐
11	Nasi Bakar	195.00	1.80	2.50	48.00	1500.00	☐
12	Nasi Liwet	365.00	7.13	0.66	79.85	2350.00	☒
13	Buras	88.00	2.30	1.30	16.70	750.00	☐
14	Sereal Beras	432.00	5.43	12.90	73.76	5000.00	☐
15	Nasi Kuning	475.00	7.80	15.80	80.00	3000.00	☐
16	Bubur kacang Hijau	205.00	5.00	3.20	45.00	2000.00	☐

Gambar 5.10 Tampilan halaman pilih makanan.

5.3.8 Halaman nutrisius

Halaman nutrisius adalah halaman output yang menampilkan berapakah gram makanan terpilih untuk dikonsumsi sesuai dengan kebutuhan energi dari individu beserta besar biayanya. Sistem juga menampilkan langkah-langkah

perhitungan penyelesaian masalah dengan metode Simpleks Big-M tersebut. Implementasi halaman nutriuser dapat dilihat pada Gambar 5.11 sampai dengan 5.13.



No	Fungsi Tujuan
1	$Z = 2300.00x_1 + 3500.00x_2 + 1500.00x_3 + 3000.00x_4 + 2350.00x_5$

No	Fungsi Kendala
1.	$2.10x_1 + 5.40x_2 + 4.10x_3 + 6.30x_4 + 7.13x_5 \leq 173.260815$
2.	$0.10x_1 + 0.50x_2 + 1.30x_3 + 6.23x_4 + 0.66x_5 \leq 77.004806666667$
3.	$40.60x_1 + 40.00x_2 + 30.30x_3 + 21.06x_4 + 79.85x_5 \leq 321.770085$
4.	$2.10x_1 + 5.40x_2 + 4.10x_3 + 6.30x_4 + 7.13x_5 \geq 49.50309$
5.	$0.10x_1 + 0.50x_2 + 1.30x_3 + 6.23x_4 + 0.66x_5 \geq 44.002746666667$
6.	$40.60x_1 + 40.00x_2 + 30.30x_3 + 21.06x_4 + 79.85x_5 \geq 222.763905$
7.	$178.00x_1 + 150.00x_2 + 149.00x_3 + 250.00x_4 + 365.00x_5 = 1980.1236$
8.	$x_1 \geq 1$
9.	$x_2 \geq 1$

No	Fungsi Tujuan
1	$Z = 2300.00x_1 + 3500.00x_2 + 1500.00x_3 + 3000.00x_4 + 2350.00x_5 + MA_1 + MA_2 + MA_3 + MA_4 + MA_5 + MA_6 + MA_7 + MA_8 + MA_9$

No	Model Standar
1.	$2.10x_1 + 5.40x_2 + 4.10x_3 + 6.30x_4 + 7.13x_5 + S_0 = 173.260815$
2.	$0.10x_1 + 0.50x_2 + 1.30x_3 + 6.23x_4 + 0.66x_5 + S_1 = 77.004806666667$
3.	$40.60x_1 + 40.00x_2 + 30.30x_3 + 21.06x_4 + 79.85x_5 + S_2 = 321.770085$
4.	$2.10x_1 + 5.40x_2 + 4.10x_3 + 6.30x_4 + 7.13x_5 - S_3 + A_1 = 49.50309$
5.	$0.10x_1 + 0.50x_2 + 1.30x_3 + 6.23x_4 + 0.66x_5 - S_4 + A_2 = 44.002746666667$
6.	$40.60x_1 + 40.00x_2 + 30.30x_3 + 21.06x_4 + 79.85x_5 - S_5 + A_3 = 222.763905$
7.	$178.00x_1 + 150.00x_2 + 149.00x_3 + 250.00x_4 + 365.00x_5 + A_4 = 1980.1236$
8.	$x_1 - S_7 + A_5 = 1$

Gambar 5.11 Tampilan halaman nutriuser (pemrograman linear).

DIET SIMPLEKS

localhost/dietsimpleks/user/nutrisiuser

START HAKIM10

PROFIL LOG OUT

MATRIKS AKHIR (Iterasi ke- 12)

Kolom	Variabel	x1	x2	x3	x4	x5	S1	S2	S3	S4	A1	S5	A2	S6	A3	A4	S7	A5	S8	A6
Cj	Cj	2300.00	3500.00	1500.00	3000.00	2350.00	0	0	0	0	10000	0	10000	0	10000	10000	0	10000	0	1000
0	S1	-3.901	0	0	0	-4.975	1	0	0	0	0	0	0	0.059	-0.059	-0.02	0	0	0	0
0	S2	-1.306	0	0	0	-2.625	0	1	0	0	0	0	0	-0.118	0.118	-0.035	0	0	0	0
0	S3	0	0	0	0	0	0	0	1	0	0	0	0	1	-1	0	0	0	0	0
10000	A9	0	0	0	0	1	0	0	0	0	0	0	0	-0	0	-0	0	0	0	0
10000	A2	-1.306	0	0	0	-2.625	0	0	0	0	0	-1	1	-0.118	0.118	-0.035	0	0	0	0
3500.00	x2	0.936	1	0	0	1.794	0	0	0	0	0	0	0	-0.037	0.037	-0.003	0	0	0	0
0	S10	0.151	0	0	0	0.383	0	0	0	0	0	0	0	0.022	-0.022	0.006	0	0	0	0
10000	A5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0
0	S8	0.936	0	0	0	1.794	0	0	0	0	0	0	0	-0.037	0.037	-0.003	0	0	1	-1
0	S4	3.901	0	0	0	4.975	0	0	0	1	-1	0	0	-0.059	0.059	0.02	0	0	0	0
3000.00	x4	0.151	0	0	1	0.383	0	0	0	0	0	0	0	0.022	-0.022	0.006	0	0	0	0
1500.00	x3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 5.12 Tampilan halaman nutriuser (tabel matriks).

DIET SIMPLEKS

localhost/dietsimpleks/user/nutrisiuser

START HAKIM10

PROFIL LOG OUT

TABEL RESULT

No	Variabel	Nama Makanan	Jumlah
1	x1	Nasi Putih	0
2	x2	Nasi Beras Hitam	1.3963621157725
3	x3	Nasi Jagung	1
4	x4	Nasi Goreng	6.4866771305365
5	x5	Nasi Liwet	0

Copyright © AHP 2015

Gambar 5.13 Tampilan halaman nutriuser (tabel result).

BAB 6 PENGUJIAN DAN ANALISIS

6.1 Sistematika pengujian

Seperti pada penjelasan bab 4 mengenai perancangan uji coba dan analisis, maka pada sistematika pengujian ini akan dilakukan beberapa macam uji coba. Sistematika pengujian tersebut adalah sebagai berikut :

6.1.1 Sistematika pengujian validitas hasil perhitungan

Pengujian pertama adalah menguji validitas hasil perhitungan, dimana dilakukan perhitungan manual pada suatu output makanan. Perhitungan manual ini dilakukan dengan cara mensubstitusikan nilai hasil keluaran ke dalam fungsi kendala dan fungsi tujuan yang telah dibuat sebelumnya.

6.1.2 Sistematika pengujian sensitivitas nilai minimal konsumsi makanan

Pengujian kedua dilakukan dengan cara mengubah nilai minimal konsumsi makanan yang digunakan sebagai batasan dalam fungsi kendala. Hal ini digunakan untuk mencari total biaya yang paling optimum.

6.1.3 Sistematika pengujian sensitivitas jenis dan jumlah pilihan makanan

Pengujian ketiga dilakukan dengan cara mengubah jumlah input pilihan makanan dalam tiga kategori yaitu makanan yang dipilih secara random, makanan dipilih berdasarkan harga terendah dan makanan dipilih berdasarkan harga tertinggi. Hal ini digunakan untuk mengetahui adakah pengaruh jumlah makanan pada beberapa kasus tertentu dalam menentukan hasil keluaran yang optimal.

6.2 Analisa dan hasil pembahasan

6.2.1 Hasil pengujian validitas hasil perhitungan

Pada pengujian perhitungan ini disubstitusikan hasil perhitungan ke dalam fungsi kendala. Pengujian ini dilakukan dengan menginputkan makanan yang ingin dikonsumsi yang kemudian di proses oleh sistem. Sebagai kasus pengujian dapat dilihat pada Gambar 6.1 dan Gambar 6.2 untuk proses pembentukan fungsi tujuan dan fungsi kendala.

KEBUTUHAN USER				
	Kalori	Protein	Lemak	Karbohidrat
Batas Bawah	1980.1236	49.50309	44.002746666667	222.763905
Batas atas		173.260815	77.004806666667	321.770085

MAKANAN TERPILIH						
Id Makanan	Nama Makanan	Kalori	Protein	Karbohidrat	Lemak	Harga
35	Tempe	193.00	18.54	10.80	9.39	850.00
129	mangga	107.00	0.84	0.45	28.05	1850.00
130	melon	19.00	0.46	0.10	4.49	2250.00

Gambar 6.1 Kebutuhan user dan makanan terpilih beserta nutrisinya.

FUNGSI TUJUAN	
No	Fungsi Tujuan
1	$Z = 850.00x_1 + 1850.00x_2 + 2250.00x_3$

FUNGSI KENDALA	
No	Fungsi Kendala
1.	$18.54x_1 + 0.84x_2 + 0.46x_3 \leq 173.260815$
2.	$10.80x_1 + 0.45x_2 + 0.10x_3 \leq 77.004806666667$
3.	$9.39x_1 + 28.05x_2 + 4.49x_3 \leq 321.770085$
4.	$18.54x_1 + 0.84x_2 + 0.46x_3 \geq 49.50309$
5.	$10.80x_1 + 0.45x_2 + 0.10x_3 \geq 44.002746666667$
6.	$9.39x_1 + 28.05x_2 + 4.49x_3 \geq 222.763905$
7.	$193.00x_1 + 107.00x_2 + 19.00x_3 = 1980.1236$
8.	$x_1 \geq 1$
9.	$x_2 \geq 1$
10.	$x_3 \geq 1$

Gambar 6.2 Fungsi tujuan dan fungsi kendala

Dari data-data yang sudah dimasukkan dalam proses perhitungan kemudian didapatkan kuantitas makanan sesuai dengan Gambar 6.3 berikut.

TABEL RESULT			
No	Variabel	Nama Makanan	Jumlah
1	x1	Tempe	6.8737391282651 satuan (1031.0608692398 gram)
2	x2	mangga	5.9298312920078 satuan (889.47469380117 gram)
3	x3	melon	1 satuan (150 gram)
TOTAL HARGA			19062.86614924 Rupiah

Gambar 6.3 Hasil perhitungan

Berdasarkan hasil perhitungan yang dilakukan oleh sistem pada Gambar 6.3 kemudian hasil tersebut coba disubstitusikan ke dalam fungsi kendala dan fungsi tujuan yang telah terbentuk sebelumnya. Dari pengecekan ini dapat dilihat apakah data yang dihasilkan sudah benar dan sesuai.

Berdasarkan output pada Gambar 6.3 didapatkan :

- Tempe = 6,873739128 satuan (x1)
- Mangga = 5,929831292 satuan (x2)
- Melon = 1 satuan (x3)
- Harga = 19062,86615 Rupiah

a. Validasi untuk fungsi kendala protein

Adapun fungsi kendala untuk protein adalah sebagai berikut :

- $18,54x_1 + 0,84x_2 + 0,46x_3 \geq 49,50309$
- $18,54x_1 + 0,84x_2 + 0,46x_3 \leq 173,260815$

Maka, dari hasil keluaran yang didapat jika disubstitusikan ke salah satu fungsi tersebut menghasilkan hasil berikut :

$$18,54(6,873739128) + 0,84(5,929831292) + 0,46(1) \geq 49,50309$$

$$132,8801817 \geq 49,50309 \quad (\text{BENAR})$$

$$132,8801817 \leq 173,260815 \quad (\text{BENAR})$$

b. Validasi untuk fungsi kendala lemak

Adapun fungsi kendala untuk lemak adalah sebagai berikut :

- $10,8x_1 + 0,45x_2 + 0,1x_3 \geq 44,00274667$
- $10,8x_1 + 0,45x_2 + 0,1x_3 \leq 77,00480667$

Maka, dari hasil keluaran yang didapat jika disubstitusikan ke salah satu fungsi tersebut menghasilkan hasil berikut :

$$10,8(6,873739128) + 0,45(5,929831292) + 0,1(1) \geq 44,00274667$$

$$77,00480667 \geq 49,50309 \quad (\text{BENAR})$$

$$77,00480667 \leq 77,00480667 \quad (\text{BENAR})$$

c. Validasi untuk fungsi kendala karbohidrat

Adapun fungsi kendala untuk karbohidrat adalah sebagai berikut :

- $9,39x_1 + 28,05x_2 + 4,49x_3 \geq 222,763905$
- $9,39x_1 + 28,05x_2 + 4,49x_3 \leq 321,770085$

Maka, dari hasil keluaran yang didapat jika disubstitusikan ke salah satu fungsi tersebut menghasilkan hasil berikut :

$$9,39(6,873739128) + 28,05(5,929831292) + 4,49(1) \geq 222,763905$$

$$235,3661782 \geq 222,763905 \quad (\text{BENAR})$$

$$235,3661782 \leq 321,770085 \quad (\text{BENAR})$$

d. Validasi untuk fungsi kendala Kalori

Adapun fungsi kendala untuk kalori adalah sebagai berikut :

$$- 193x_1 + 107x_2 + 19x_3 = 1980,1236$$

Maka, dari hasil keluaran yang didapat jika disubstitusikan ke dalam fungsi tersebut menghasilkan hasil berikut :

$$193(6,873739128) + 107(5,929831292) + 19(1) \geq 1980,1236$$

$$235,3661782 = 1980,1236 \quad (\text{BENAR})$$

e. Validasi untuk fungsi tujuan

Adapun fungsi tujuan adalah sebagai berikut :

$$- Z = 850 x_1 + 1850x_2 + 2250x_3$$

Maka, dari hasil keluaran yang didapat jika disubstitusikan ke dalam fungsi tersebut menghasilkan hasil berikut :

$$850 (6,873739128) + 1850 (5,929831292) + 2250 (1) = 19062,86615$$

$$19062,86615 = 19062,86615 \quad (\text{BENAR})$$

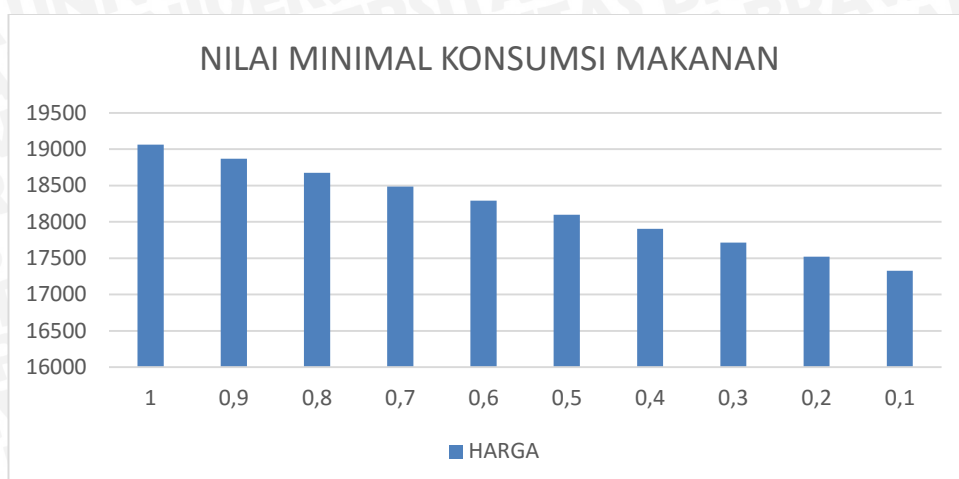
Dari Kelima proses substitusi dapat disimpulkan bahwa hasil perhitungan yang disubstitusikan ke dalam fungsi batasan maupun fungsi tujuan ternyata bernilai valid dan benar. Namun hasil perhitungan tersebut perlu diuji kembali apakah hasil yang didapat benar-benar optimal. Untuk mengetahui optimal atau tidaknya maka perlu dilakukan pengujian kembali terhadap hasil-hasil perhitungan tersebut yaitu dengan cara pengujian optimalitas.

6.2.2 Hasil pengujian sensitivitas nilai minimal konsumsi makanan

Pada pengujian ini akan dilakukan perubahan pada nilai minimal konsumsi makanan yang dalam sistem merupakan nilai batas pada fungsi kendala secara manual. Pada skenario pengujian ini, nilai batas akan diuji ke beberapa titik, mulai dari 1 dan akan dikurangkan nilainya sejumlah 0,1 pada uji percobaan berikutnya. Hasil uji coba pengujian ini dapat ditunjukkan pada Tabel 6.1.

Tabel 6.1 Hasil pengujian sensitivitas nilai minimal konsumsi makanan

Batas	Makanan			Harga	Total Biaya
	X1	X2	X3		
1	6,874	5,93	1	X1 = 850 ; X2 = 1850 ; X3 = 2250 ;	19062,866
0,9	6,874	5,947	0,9		18870,216
0,8	6,874	5,965	0,8		18677,567
0,7	6,874	5,982	0,7		18484,917
0,6	6,875	5,999	0,6		18292,267
0,5	6,875	6,017	0,5		18099,617
0,4	6,875	6,034	0,4		17906,967
0,3	6,875	6,052	0,3		17714,317
0,2	6,875	6,069	0,2		17521,668
0,1	6,876	6,086	0,1		17329,018



Gambar 6.4 Grafik hasil pengujian nilai minimal konsumsi makanan.

Dilihat dari Tabel 6.1 dan Gambar 6.4 dapat dianalisa bahwa nilai batas / nilai minimal konsumsi makanan yang paling tepat untuk mendapatkan total biaya pengeluaran terkecil adalah pada batas kendala 0,1(dengan asumsi per 150 gr) dengan total pengeluaran sebesar 17329,018 Rupiah. Hal ini juga membuktikan bahwa dengan mengurangi nilai batas konsumsinya tidak mengurangi nilai kecukupan gizi individu tetapi dapat menghasilkan biaya yang lebih kecil.

6.2.3 Hasil pengujian sensitivitas jenis dan jumlah pilihan makanan

Pada pengujian ini dilakukan uji jumlah makanan ke beberapa nilai. Pada skenario pengujian ini, jumlah makanan yang diuji ada 6 varian yaitu 1,2,3,5,7 dan 9. Adapun kategori pengujian dibagi menjadi 3 yaitu makanan yang dipilih secara random, makanan dipilih berdasarkan harga terendah dan makanan dipilih berdasarkan harga tertinggi. Hasil uji coba pengujian ini dapat ditunjukkan pada Tabel 6.2, 6.3 dan 6.4 (data makanan ditunjukkan pada Lampiran Data daftar makanan).

Tabel 6.2 Hasil uji 1 – makanan dipilih secara random

jumlah makanan	id makanan	validitas					optimalitas
		kalori	protein	lemak	karbohidrat	harga	
1	94	x	x(DD)	v	x	x(SB)	x
2	94, 141	x(DD)	v	v	x(DD)	v	v
3	94,141,87	v	v	v	v	v	v
5	94,141,87,132,57	v	x(DD)	v	v	x(B)	x
7	94,141,87,132,57,7,166	x(DD)	v	v	v	x(B)	x
9	94,141,87,132,57,7,166,155,32	x(DD)	v	v	v	x(B)	x

Tabel 6.3 Hasil uji 2 – makanan dipilih berdasarkan harga terendah

jumlah makanan	id makanan	validitas					optimalitas
		kalori	protein	lemak	karbohidrat	harga	
1	132	x(DD)	v	v	x	x(SB)	x
2	132, 13	x(DD)	v	x(DD)	v	v	v
3	132,13,20	x(DD)	v	v	v	v	v
5	132,13,20,116,35	v	v	v	v	v	v
7	132,13,20,116,35,131,135	x(DD)	x(DD)	v	v	x(K)	x
9	132,13,20,116,35,131,135,74,141	v	v	x(DD)	v	x(K)	x

Tabel 6.4 Hasil uji 3 – makanan dipilih berdasarkan harga tertinggi

jumlah makanan	id makanan	validitas					Optimalitas
		kalori	protein	lemak	karbohidrat	harga	
1	57	x	v	x	x	x(SB)	x
2	57, 41	x	v	x	x	x(SB)	x
3	57,41,59	x	v	x	x	x(SB)	x
5	57,41,59,94,58	x	v	x	x	x(SB)	x
7	57,41,59,94,58,55,81	x(DD)	x(DD)	x(DD)	x	x(SB)	x
9	57,41,59,94,58,55,81,52,87	x(DD)	v	v	x	x(SB)	x

Tabel 6.5 Keterangan indeks pengujian

indeks	keterangan
v	memenuhi
x	tidak memenuhi
SB (sangat besar)	ketidaksesuaian validitas harga >100000
B (besar)	ketidaksesuaian validitas harga >10000 & <100000
S (sedang)	ketidaksesuaian validitas harga >1000 & <10000
K (kecil)	ketidaksesuaian validitas harga >1 & <1000
DD (<i>decrease decimal</i>)	ketidaksesuaian validitas nutrisi tidak memenuhi < 0,1

Dilihat dari Tabel 6.2 sampai Tabel 6.4, dalam beberapa kasus ujicoba dimana terdapat 3 kategori pengujian dan masing-masing kategori terdapat 6 pengujian berdasarkan jumlah input makanannya. Dari hasil pengujian tersebut dapat dianalisa sebagai berikut.

1. Metode Simpleks Big-M mendapati ketidaksesuaian validitas nutrisi maupun harga yang sangat tinggi apabila jumlah input makanan berjumlah 1. Jumlah nilai M mempengaruhi nilai keluaran dikarenakan masih adanya variabel *Artificial* pada Tabel Simpleks hingga iterasi terakhir. Sehingga membuat keluaran harga menjadi sangat besar. Selain itu metode Simpleks Big-M adalah metode yang digunakan untuk memecahkan permasalahan pemrograman linear dengan dua variabel atau lebih.
2. Dalam beberapa kasus ujicoba pada Tabel 6.2 dapat dianalisa bahwa tingkat validitas untuk gizi (kalori, protein, lemak dan karbohidrat) cukup tinggi. Sedangkan ketidaksesuaian besar untuk harga dengan jumlah makanan 5,

7 dan 9. Adapun tingkat validitas untuk gizi terdapat beberapa nilai keluaran yang hanya berbeda nilai decimalnya saja (dalam Tabel 6.2 ditunjukkan dengan indeks $x(DD)$). Ketidaksesuaian ini kemudian merupakan nilai yang dapat ditolerir karena perbedaannya yang kecil dan tidak cukup berpengaruh dalam pemenuhan gizi individu.

3. Dalam beberapa kasus ujicoba pada Tabel 6.3 dapat dianalisa bahwa tingkat validitas untuk gizi (kalori, protein, lemak dan karbohidrat) cukup tinggi. Sedangkan ketidaksesuaian pada harga dengan jumlah makanan 7 dan 9 sangat kecil (dalam Tabel 6.3 ditunjukkan dengan indeks $x(K)$). Adapun tingkat validitas untuk gizi terdapat beberapa nilai keluaran yang hanya berbeda nilai decimalnya saja (dalam Tabel 6.3 ditunjukkan dengan indeks $x(DD)$). Ketidaksesuaian ini kemudian merupakan nilai yang dapat ditolerir karena perbedaannya yang kecil dan tidak cukup berpengaruh dalam pemenuhan gizi individu. Sedangkan untuk perbedaan harga antara sistem dan uji validitasnya, adalah sesuatu yang wajar karena metode Simpleks harus mencari hasil yang paling optimal dari fungsi kendala dan fungsi tujuan sehingga terdapat beberapa nilai yang tidak berada dalam range batasannya. Dalam pengujian ini dapat disimpulkan bahwa uji makanan dengan harga yang relatif murah tidak mempengaruhi hasil keluaran yang optimal secara signifikan.
4. Dalam beberapa kasus ujicoba pada Tabel 6.4 dapat dianalisa bahwa tingkat validitas untuk gizi (kalori, protein, lemak dan karbohidrat) maupun harga sangat buruk. Perbedaan hasil uji validitas dan hasil keluaran baik gizi maupun harga sangat besar (dalam Tabel 6.4 ditunjukkan dengan indeks $x(SB)$). Hal ini dapat dipengaruhi karena adanya perbedaan nilai gizi maupun harga yang sangat besar. Perbedaan ini kemudian membuat masalah dalam fungsi kendala dan fungsi tujuan dikarenakan tidak terdapat daerah *feasible* (*no feasible solution*). Tidak adanya daerah *feasible* ini mengakibatkan tidak adanya solusi optimal. Selain itu *error* harga yang sangat besar diakibatkan karena nilai M yang ditentukan untuk menghitung solusi sangat besar.

Sehingga dari pengujian 6.2.4 ini dapat disimpulkan bahwa jumlah makanan yang dikonsumsi dan jenis makanan yang dikonsumsi sangat mempengaruhi tingkat validitas dan optimalitas dalam penyelesaian masalah optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa dengan menggunakan metode Simpleks Big-M. Tiap makanan yang dipilih akan menentukan adanya solusi optimal dikarenakan kandungan gizi dan harga merupakan penyusun dalam membuat fungsi kendala dan fungsi tujuan. Tidak adanya solusi optimal dipengaruhi karena tidak adanya daerah *feasible* atau berada pada *unbounded objective function*.

BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil uji coba dan analisis yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut.

1. Metode Simpleks Big-M dapat digunakan untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa. Metode Simpleks Big-M dapat menentukan kuantitas makanan yang dipilih beserta total biaya yang harus dikeluarkan oleh individu secara optimal.
2. Dengan membuat nilai minimal konsumsi makanan semakin kecil, maka akan membuat harga semakin optimal tanpa mengurangi nilai kebutuhan gizi individu.
3. Jenis dan jumlah makanan yang dikonsumsi dapat mempengaruhi keoptimalan metode Simpleks Big-M untuk menyelesaikan permasalahan optimasi biaya pemenuhan gizi dan nutrisi pada orang dewasa. Metode Simpleks Big-M tidak cocok untuk digunakan jika individu hanya mengkonsumsi satu jenis makanan saja. Selain itu Metode Simpleks Big-M juga kurang dapat merekomendasikan gizi dan harga secara optimal apabila semua makanan yang dipilih adalah makanan dengan harga yang sangat mahal.

7.2 Saran

Dalam penelitian ini masih banyak hal-hal yang dapat dikembangkan dan dijadikan bahan pertimbangan. Adapun saran untuk pengembangan penelitian lebih lanjut adalah :

1. Dalam beberapa kasus permasalahan tertentu, sebab adanya perbedaan nilai dibelakang koma yang nilainya sangat kecil mengakibatkan ketidaksesuaian dalam menguji kevalidannya, untuk itu pembulatan nilai ke beberapa angka dibelakang koma perlu dilakukan. Metode *Integer programming* bisa menjadi alternatif dalam penyelesaian masalah ini.
2. Pada penelitian kali ini masih menggunakan jenis data yang sedikit sehingga perlu ditambahkan berbagai varian makanan.
3. Harga yang diperoleh pada penelitian ini menggunakan data *dummy* yang disesuaikan dengan kondisi permasalahan dan harga bahan makanan dapat berubah sewaktu-waktu jadi diharuskan adanya *update* data.
4. Untuk mengatasi permasalahan makanan dengan harga yang sangat tinggi sedangkan gizi nya sangat rendah. Salah satu cara untuk mengatasi permasalahan pilihan menu makanan dengan harga mahal dan gizi rendah adalah dengan menambahkan pilihan makanan lainnya dalam pilihan menu.

DAFTAR PUSTAKA

- A.Ravindan, T.Philips, D. & S.Sorlberg, J., 1987. *Operations research prnciples and practice*. 2nd ed. US: John Wiley & son,Inc.
- A.Taha, H., 2007. *Operations research: an introduction*. 8th ed. Upper Saddle River: Pearson Education,Inc.
- Almatsier, S., 2001. *Prinsip ilmu gizi*. Jakarta: PT. Gramedia Pustaka Utama.
- Almatsier, S., 2005. *Penuntun Diet*. Jakarta: PT. Gramedia Pustaka Utama.
- Food and Nutrition Board, I. M. N. A., 2005. *Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids*. [Online] Tersedia di: <https://fnic.nal.usda.gov/dietary-guidance> [Diakses 25 September 2015].
- Kesehatan, B. P. d. P., 2013. *Riset Kesehatan Dasar 2013*, Kementerian Kesehatan.
- Levin, R. I., 2012. *Pengambilan Keputusan Secara Kuantatif*. 7th ed. Jakarta: PT Raja Grafindo Persada.
- McAuley, D., 1993. *Revised Harris Benedict equation - Determination of the (BMR)*. [Online] Tersedia di: <http://www.globalrph.com/revised-harris-benedict-equation.htm> [Diakses 25 September 2015].
- Muslich, M., 2010. *Metode Pengambilan Keputusan Kuantitatif*. 1st ed. Jakarta: Bumi Aksara.
- Oruç, K. O., Güngör, I. & Sezgin Irmak2, S. Ş., 2012. *Menu Planning with Fuzzy 0-1 Integer Programming*. Sarajevo,
- Santoso, R. A., 2013. *Desain dan Implementasi Sistem Informasi Pengelolaan Administrasi Biaya Proyek Konstruksi Bangunan dengan Metode Simpleks*, S1: Universitas Brawijaya.
- Suci, W. W., 2015. *Optimasi Biaya Pemenuhan Gizi dan Nutrisi pada Manusia Lanjut Usia Menggunakan Algoritma Genetika*, S1: Universitas Brawijaya.
- Wahyuningsih, R., 2013. *Penatalaksanaan diet pada pasien*. Yogyakarta: Graha Ilmu.
- Wijaya, A., 2013. *Pengantar Riset Operasi*. 3rd ed. Jakarta: Mitra Wacana Media.

LAMPIRAN DATA DAFTAR MAKANAN

Lampiran 1 Data daftar makanan satuan/150 gr (Depkes 2010 & Penelitian (Suci, 2015))

ID	NAMA MAKANAN	KALORI (kal)	PROTEIN (gram)	LEMAK (gram)	KARBOHIDRAT (gram)	HARGA (Rp)
1	Nasi Putih	178	2.1	0.1	40.6	2300
2	Nasi Beras Merah	179.5	3.75	0.45	38.5	2750
3	Nasi Beras Hitam	150	5.4	0.5	40	3500
4	Nasi Jagung	149	4.1	1.3	30.3	1500
5	Papeda	84.8	61	0.2	14.9	1750
6	Nasi Uduk	232	3	12	28	3500
7	Nasi Tim daging	250	11	15.4	56.1	4000
8	nasi minyak	265	15	17	62	4500
9	Nasi Goreng	250	6.3	6.23	21.06	3000
10	Ketupat	182	1.9	0.1	42	1000
11	Nasi Bakar	195	1.8	2.5	48	1500
12	Nasi Liwet	365	7.13	0.66	79.85	2350
13	Buras	88	2.3	1.3	16.7	750
14	Sereal Beras	432	5.43	12.9	73.76	5000
15	Nasi Kuning	475	7.8	15.8	80	3000
16	Bubur kacang Hijau	205	5	3.2	45	2000
17	Bubur Sumsum	170	3.8	2.1	34	1000
18	Bubur Tinotuan	350	19	8.3	34.7	2500
19	bubur ayam	372	27.5	12.3	36.12	1600
20	Pulut	216	2.6	3.1	44.4	750
21	Ketan Hitam	169	3.51	0.33	36.7	1800
22	Ketan putih	178	3	0.33	37	1850
23	Roti tawar bakar	264	9.6	3.2	48.9	1400
24	Roti gandum	285	10	4	52	1600
25	bubur jagung	91.5	2.56	0.34	23.1	1500
26	oatmeal	130	3	0	28	2000
27	bubur wortel	7.3	1.4	0	56	1500
28	bubur kacang merah	106	3.9	3.6	53.2	3000
29	Bubur Ketan Hitam	189	5.01	2	40	2500
30	Bubur Manado	300	38.9	15	69	3000
31	Tahu Goreng	271	17.19	20.18	10.49	1000
32	pepes tahu	126	11.42	7.11	4.29	1950
33	Sapo tahu	150	19.9	6.6	2.7	3500
34	Tahu isi Ayam	112	8.54	5.53	7.72	1750

35	Tempe	193	18.54	10.8	9.39	850
36	Ayam Panggang	147	16.7	8.36	0	2950
37	Ayam Kecap	150	15	9.6	0	2750
38	Sop Ayam	75	1.05	2.46	9.36	2600
39	Gulai Ikan	126	4.2	9.2	6.2	6000
40	Bakmie Ayam	356	35	12.5	30	3500
41	Dendeng Sapi	476	45	9.8	4.2	18000
42	Burung Puyuh	131	20.2	4.3	4.8	5000
43	dendeng Itik	496	30.3	33.5	18.4	7500
44	Ayam Taliwang	264	18.2	20.1	2.7	8500
45	Tenggiri	109	21.5	2.6	0	4500
46	Gudeg	345	40.5	12	5	3500
47	Ikan Belida Bakar	128	18	3	7.2	6000
48	Ikan Cakalang	107	19.6	0.7	5.5	7500
49	Ikan baronang	78	14.5	0.6	3.7	4500
50	Gulai Telur Ikan	146	12.3	7.1	8.1	6000
51	Sate Ayam	225	19.54	14.82	1.89	1000
52	Abon Ikan	435	27.2	20.2	36.1	8750
53	Ikan patin bakar	144	17.5	6.3	4.3	5750
54	gulai Ikan paya	148	10	9.8	5	7500
55	coto makasar	84	6	6	1.4	10500
56	Teri	74	10.3	1.4	4.1	6500
57	Sidat	81	11.4	1.9	3.8	21500
58	Tuna	92	20	0.7	0	10750
59	Hati sapi	132	19.7	3.2	6	15000
60	abon sapi	212	18	10.6	59.3	5650
61	pindang	124	9.5	9.6	0	4500
62	bandeng presto	296	17.1	20.3	11.3	7750
63	soto ayam	312	24.01	14.92	19.55	2500
64	telur ayam	162	12.8	11.5	0.7	1500
65	telur bebek	189	13.1	14.5	0.8	1700
66	telur puyuh	168	12.3	12.7	1.2	1800
67	ikan lele	204	14.93	12.35	12.35	1750
68	ikan kembung	167	19.23	9.3	0	1600
69	ikan nila	128	26.15	2.65	0	3450
70	ikan tuna sirip biru	144	23.33	4.9	0	6575
71	ikan kakap	100	20.51	1.34	0	7800
72	ikan salmon	183	19.9	10.85	0	2500
73	ikan tongkol	110	23.87	0.92	0	2000
74	Bihun goreng	192	1.6	0.35	43.82	1000
75	ayam kecap	421	16.7	18.74	46.21	1850
76	mie telur goreng	146	5.38	1.69	27.08	2000
77	telur orak arik	199	13.01	15.21	1.96	2150

78	telur balado	202	10.21	16.43	3.4	2550
79	kari ayam	124	11.47	6.67	4.74	3250
80	kari daging sapi	184	11.69	12.96	2.8	8000
81	rendang	195	19.68	11.07	4.49	9000
82	daging kornet	251	18.71	18.89	0.47	6750
83	sosis ayam	172	17.82	9.98	1.52	2000
84	sosis sapi	325	11.04	29.8	3.99	3500
85	dimsum	112	11.55	2.64	9.56	1000
86	rawon daging	119	9.6	7.4	3.48	6750
87	mie china	237	3.77	13.84	25.89	8750
88	bothok	336	23.4	3.2	59.9	1000
89	gulai telur ikan	146	12.3	7.1	8.1	6000
90	oncom	187	13	6	22.6	2350
91	kembang tahu	380	48.9	13.8	23.3	4000
92	bawal goreng	96	4.16	1.7	2	5000
93	soto banjar	110	2.9	9.5	3.2	4500
94	sop konro	71	7.4	2.6	4.5	11850
95	Sayur Bayam	74	5.33	4.16	6.81	1700
96	Brokoli	20	3.5	0.1	3.1	3200
97	perkedel kentang	83	2	0.1	19.1	1350
98	Macaroni Panggang	365	8.7	0.4	78.7	4500
99	Sup Kembang Tahu	380	48.9	13.8	23.3	4000
100	Cah Jamur Kuping	21	3.8	0.6	0.9	2700
101	Capjay	189	11.3	15.7	0.6	5000
102	Sayur asem	85	1.1	0.8	0.7	2300
103	kembang kol	22	2.8	0.5	1.12	3450
104	sup kubis	71	3.92	3.26	7.69	2650
105	sop brokoli	20	3.5	0.1	3.1	3200
106	bayam merah	5.1	4.6	0.5	10	2150
107	oseng daun singkong	60	1.1	7.1	2.4	1450
108	oseng kangkung	92	2.2	9.2	2	1700
109	sayur lodeh	104	3.48	2.62	17.44	2000
110	urap	112	2.32	7.65	10.89	1500
111	pecel	463	15.63	30.38	40.04	2350
112	sayur sop	90	10.7	4	4.7	2000
113	sayur buntel	1441	4.1	10.2	8	1750
114	gado-gado	203	6.7	8.7	24.6	2500
115	sayur tumis toge	114	3.4	5	0	1000
116	tumis bayam	193	2	0	0	850
117	sayur telur wortel	140	10.7	0.5	0	3500
118	cah brokoli	249	12.45	7	5	3750
119	tumis teri buncis	158	33.5	3	5	2000

120	Alpukat	322	1.02	29.47	17.45	3000
121	apel	72	0.36	0.23	19.06	1500
122	jambu biji	49	0.9	0.2	12.2	1750
123	jeruk manis	45	0.9	0.2	12.4	3500
124	semangka	30	0.61	0.15	7.55	3250
125	jus strawberry	32	0.67	0.23	7.68	1650
126	pepaya	55	0.85	0.2	13.37	3000
127	jeruk mandarin	37	0.57	0.22	9.34	3750
128	pisang ambon	99	1.2	0.2	20.9	2000
129	mangga	107	0.84	0.45	28.05	1850
130	melon	19	0.46	0.1	4.49	2250
131	susu sapi	122	8.03	4.88	11.49	950
132	susu kedelai	54	4.69	1.99	5.1	500
133	pisang susu	118	1.2	0.2	31.8	1850
134	pisang cavandis	150	6	0.2	40.8	3000
135	Ubi Kuning	72.6	119	0.5	25.1	1000
136	Singkong Rebus	77.8	88	0.4	20.6	1300
137	Onde- Onde	336	4	57.8	57.8	2000
138	Apem	148	2	0.5	33.9	1500
139	Rangginang	471	4.7	21.8	64.1	3500
140	Bingka Ambon	273	53	10.6	39.1	1750
141	Bolu Pecah	197	3.3	4.6	35.6	1000
142	Putu Mayang	121	1.7	3.4	18.1	1000
143	Srikaya Ketan	265	2.7	6.4	49.1	2500
144	Kue Putu Singkong	217	1.2	0.5	53.2	2000
145	Kue Kelapa	591	56.5	42.1	47.5	3000
146	Biskuit	458	6.9	14.4	75.1	3000
147	Roti Kukus	249	5.1	2.1	52.5	3500
148	Bakpao	239	12.2	2.6	41.6	3000
149	Pastel	208	5.2	15.4	31.4	2000
150	Risoles	134	2.1	1.4	28.8	2500
151	Kelepon	215	3.7	3.7	41.8	2000
152	Kue Bakpia	272	3.7	6.7	44.1	3500
153	Nagasari	166	6.3	2.8	32.9	2500
154	Kue Lumpur	591	5.6	42.1	47.5	3500
155	Pandan Cake	86	1	5	9.6	2300
156	Brownies	129	1.62	4.68	21.26	3200
157	Donat	133	1.4	8.68	13.44	2500
158	Kue Coklat	235	2.26	10.5	34.68	3750
159	Kue Sus	182	4.58	10.92	16.58	1800
160	Ubi Madu	91	21	0.2	27	2000
161	Kue keju	257	4	18	20.4	4750
162	Puding Coklat	157	3.05	4.52	25.99	2950

163	Puding Vanilla	147	2.6	4.07	24.48	2500
164	Puding Tapioka	134	2.26	21.92	4.18	2000
165	Kue Beras Merah	35	0.75	0.25	7.34	1750
166	Roti Bakar	89	6	7.1	35	1500
167	Martabak Manis	79	5	12	33	2150
168	Lemper	95	7	6.45	55	2000
169	Orem-Orem	87	6.7	5.44	50	1500
170	Perut Ayam	76	2.7	6.7	35	1000
171	Kue Gandum	450	6.2	18.1	18	2450
172	Kroket Ayam	7.3	1.2	2.8	0.7	1750
173	Gethuk Singkong	70	2.4	1.6	11.2	1000
174	Gethuk Lindri	75	3.7	2.3	14	1500

