

IMPLEMENTASI ALGORITMA *PARTICLE SWARM* OPTIMIZATION UNTUK OPTIMASI PEMENUHAN KEBUTUHAN GIZI KELUARGA

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Felia Eliantara
NIM: 125150200111016



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

PENGESAHAN

IMPLEMENTASI ALGORITMA *PARTICLE SWARM OPTIMIZATION* UNTUK OPTIMASI
PEMENUHAN KEBUTUHAN GIZI KELUARGA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Nama Felia Eliantara
NIM: 125150200111016

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

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom
NIP. 201201 850719 1 001

Indriati, S.T, M.Kom
NIP. 19831013 201504 2 002

Mengetahui
Ketua Jurusan Teknik Informatika

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

PERNYATAAN ORISINALITAS

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

Apabila ternyata 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, 1 Januari 2015

Felia Eliantara

NIM: 125150200111016



KATA PENGANTAR

Puji syukur ke hadirat Tuhan atas berkat dan penyertaan-Nya sehingga laporan skripsi yang berjudul “Implementasi Algoritma *Particle Swarm Optimization* dalam Optimasi Pemenuhan Kebutuhan Gizi Keluarga)” bisa diselesaikan dengan baik.

Penulis menyadari benar bahwa skripsi ini tidak dapat terselesaikan dengan baik tanpa keterlibatan dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Imam Cholissodin, S.Si., M.Kom dan Ibu Indriati, S.T, M.Kom selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan dalam pengerjaan dan penulisan skripsi ini sehingga dapat terselesaikan.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Bapak Imam Cholissodin, S.Si., M.Kom selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
4. Keluarga penulis, Bapak Ir. Elianson Bungas Gerson, M.T, Ibu Dr. Ir. Evi Feronika Elbaar, M.Si, Sonia Elisurya, Margareth Karina Elishinta serta seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
5. Teman-teman penulis, Yosia Rimbo, Claudia, Nindi, Edelin, Fitria, Fitriyah, Dewi, Ando, Mahisa, Abraham, Eva, Agnes, Canny, Uly, Tiara, Dina, Desty, Olga, Rima, Ingrid dan Louisa, serta teman – teman Informatika Angkatan 2012 lainnya atas dukungan, masukan dan semangat yang diberikan kepada penulis sehingga terselesaikannya skripsi ini.
6. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 14 Agustus 2016

Felia Eliantara

feliantara@gmail.com

ABSTRAK

Menurut Peraturan Menteri Kesehatan tahun 2014 mengenai pedoman gizi seimbang, konsumsi pangan masyarakat Indonesia cenderung dibawah standar gizi seimbang. Berdasarkan Riset Kesehatan Dasar) (Riskesdas) 2013, 93,5% konsumsi sayuran dan buah-buahan pada penduduk usia di atas 10 masih di bawah anjuran. Kualitas protein yang dikonsumsi rata-rata perorang perhari juga cenderung rendah karena sebagian besar berasal dari protein nabati seperti sereal dan kacang-kacangan. Untuk konsumsi makanan dan minuman berkadar gula tinggi, garam tinggi dan lemak tinggi, baik pada masyarakat perkotaan maupun perdesaan, masih cukup tinggi. Makanan sehari-hari yang dikombinasikan dengan baik berfungsi sebagai asupan gizi yang bermanfaat banyak seperti memberi energi, membantu pertumbuhan dan pemeliharaan jaringan tubuh serta membantu jalannya proses-proses yang terjadi dalam tubuh. Hal ini tentunya juga berpengaruh besar terhadap aktifitas fisik manusia sehari-hari. Bila asupan gizi tidak tercukupi maka kemungkinan besar aktifitas pun akan terganggu. Setiap orang tentunya memiliki kebutuhan gizi yang berbeda berkaitan dengan faktor-faktor yang mempengaruhi kebutuhan asupan gizi masing-masing individu, faktor-faktor seperti usia, jenis kelamin, berat badan, tinggi badan, dan aktivitas. Maka dari itu dibutuhkan takaran bahan makanan yang tepat untuk memenuhi dengan kebutuhan gizi setiap orang. Dalam hal ini, aspek teknologi dapat membantu memberikan solusi yaitu rekomendasi kombinasi optimal bahan pangan yang dapat memenuhi kebutuhan gizi, tidak hanya satu orang tetapi kombinasi bahan makanan untuk memenuhi kebutuhan satu keluarga. Pada penelitian ini dilakukan optimasi Pemenuhan Kebutuhan gizi keluarga dengan rekomendasi kombinasi bahan makanan. Untuk proses optimasi pada penelitian ini, algoritma yang digunakan adalah *particle swarm optimization* (PSO). Kelebihan dari algoritma PSO sendiri yaitu, algoritma ini memiliki konsep sederhana, dapat diimplementasikan dengan mudah dalam perhitungan dibandingkan dengan algoritma matematika dan teknik optimisasi heuristik lainnya. Dari pengujian penyelesaian kasus actual, diketahui bahwa hasil rekomendasi dari sistem mampu memenuhi kebutuhan gizi keluarga dengan selisih yang masih berada dalam toleransi selisih kebutuhan gizi yaitu $\pm 10\%$ dan juga mampu menghemat biaya pengeluaran untuk konsumsi sebanyak 39.31%.

Kata Kunci : Gizi, bahan makanan, keluarga, makanan sehat, optimasi, *particle swarm optimization*.

ABSTRACT

According to the Ministry of Health in 2014 regarding guidelines for nutrition, food consumption rate of Indonesian society tend to be lower than the nutrition consumption standards. Based on Health Research (Riskesmas) in 2013, 93.5% of consumption of vegetables and fruits among childrens aged over 10 still under instigation. The quality of protein consumed on average per person per day also tend to be low due largely derived from plant proteins such as cereals and beans. For the consumption of foods and beverages high in sugar content, high salt and high fat diet, both in urban and rural communities, is still quite high. Good combination of daily meal that contains good nutrition to function as useful as giving a lot of energy, helps the growth and maintenance of body tissues and helps the course of the processes occurring in the body. It is of course also a major influence on human physical activity daily. When nutrition is not fulfilled then the most likely activity will be disrupted. Each person must have different nutritional needs related to factors that affect nutrition needs of each individual, factors such as age, sex, weight, height, and activity. Thus the required proper dose of the foods to meet the nutritional needs of each person. In this case, the aspect of technology can help provide a solution, that is on the optimal combination of food that can meet the nutritional needs, not just one person but the combination of food to meet the needs of a family. This research will be optimized Meeting the Needs of family nutrition combined with groceries recommendations. For process optimization in this study, the algorithm used is particle swarm optimization (PSO). The advantages of PSO algorithm itself is, this algorithm has a simple concept, it can be implemented easily in the calculation compared with a mathematical algorithm and other heuristic optimization techniques. From testing of actual cases, it is known that the recommendation of the system is able to meet the nutritional needs of families with the difference that tare within the tolerance of nutritional needs margin which is $\pm 10\%$ and is also able to save costs on consumption as much as 39.31%.

Keywords : Nutrition, food, family, healthy food, optimization, particle Swarm Optimization.

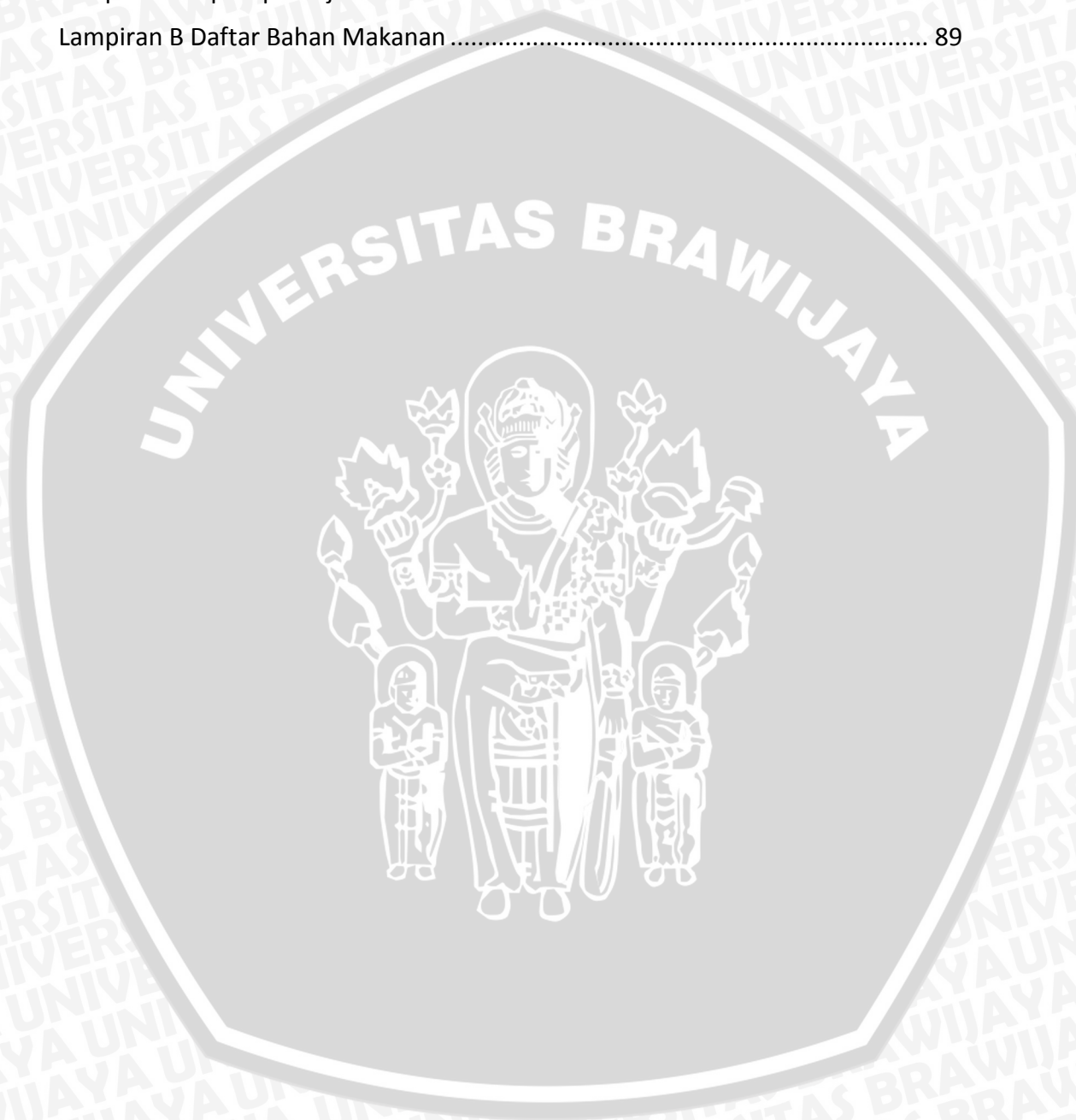
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR KODE PROGRAM	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
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 Makanan Sehat dan Gizi	9
2.3 Angka Kecukupan Gizi (AKG).....	10
2.4 Indeks Massa Tubuh (IMT).....	11
2.5 Angka Metabolisme Basal (AMB)	12
2.6 Aktivitas Fisik.....	12
2.7 Zat Gizi	13
2.7.1 Karbohidrat	13
2.7.2 Lemak	14
2.7.3 Protein.....	14
2.7.4 Energi	15
2.8 Optimasi.....	15
2.9 Algoritma <i>Particle swarm optimization</i> (PSO)	16

2.10 Nilai <i>Fitness</i>	19
BAB 3 METODOLOGI	20
3.1 Studi Literatur	20
3.2 Analisis Kebutuhan	21
3.2.1 Deskripsi Umum Sistem	21
3.2.2 Spesifikasi Kebutuhan Sistem	21
3.2.3 Analisis Kebutuhan Data	22
3.3 Perancangan Sistem.....	22
3.4 Implementasi	22
3.5 Pengujian Hasil Implementasi.....	22
3.6 Analisis Hasil dan Pembahasan.....	23
3.7 Penarikan Kesimpulan	23
BAB 4 PERANCANGAN.....	24
4.1 Penyelesaian Masalah Menggunakan Algoritma <i>Particle swarm optimization</i>	24
4.1.1 Proses inialisasi awal	26
4.1.2 Proses Perhitungan <i>Fitness</i>	27
4.1.3 Proses Perhitungan Pbest dan Gbest.....	34
4.1.4 Proses Perhitungan Perubahan Kecepatan (<i>Velocity</i>)	36
4.1.5 Proses Perhitungan Perubahan Posisi Partikel	37
4.2 Perhitungan Manual	38
4.2.1 Parameter Algoritma <i>Particle Swarm Optimization</i>	38
4.2.2 Inialisasi Partikel	38
4.2.3 Perhitungan Total Harga	41
4.2.4 Perhitungan Penalti Gizi.....	42
4.2.5 Perhitungan <i>Fitness</i>	47
4.2.6 Penentuan Pbest	48
4.2.7 Penentuan Gbest.....	48
4.2.8 Perhitungan Perubahan Kecepatan Partikel	49
4.2.9 Perhitungan Perubahan Posisi Partikel	50
4.2.10 Hasil Perhitungan Manual.....	50
4.3 Perancangan Antarmuka	53

4.4 Perancangan Uji Coba dan Evaluasi	55
4.4.1 Uji Coba Berdasarkan Banyaknya Partikel	56
4.4.2 Uji Coba berdasarkan Banyaknya Iterasi	56
4.4.3 Uji Coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}	57
4.4.4 Uji Coba Berdasarkan Batas Angka Permutasi	57
BAB 5 IMPLEMENTASI	58
5.1 Implementasi Kode Program	58
5.1.1 Inisialisasi Partikel Awal	58
5.1.2 Inisialisasi Kecepatan Awal	58
5.1.3 Perhitungan <i>Fitness</i>	59
5.1.4 Perhitungan <i>Pbest</i>	68
5.1.5 Perhitungan <i>Gbest</i>	69
5.1.6 Update Kecepatan Partikel	69
5.1.7 Update Posisi Partikel	70
5.2 Implementasi Antarmuka	70
5.2.1 Implementasi Sistem	71
BAB 6 PENGUJIAN DAN ANALISIS	74
6.1 Hasil dan Analisis Uji coba Berdasarkan Banyaknya Partikel	74
6.1.1 Hasil Pengujian	74
6.1.2 Analisis Hasil Pengujian	75
6.2 Hasil dan Analisis Uji coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}	75
6.2.1 Hasil Pengujian	76
6.2.2 Analisis Hasil Pengujian	76
6.3 Hasil dan Analisis Uji Coba Berdasarkan Banyaknya Iterasi	77
6.3.1 Hasil Pengujian	77
6.3.2 Analisis Hasil Pengujian	78
6.4 Hasil dan Analisis Uji Coba Berdasarkan Batas Angka Permutasi	78
6.4.1 Hasil Pengujian	79
6.4.2 Analisis Hasil Pengujian	79
6.5 Analisis Global Hasil Pengujian	80
BAB 7 PENUTUP	84

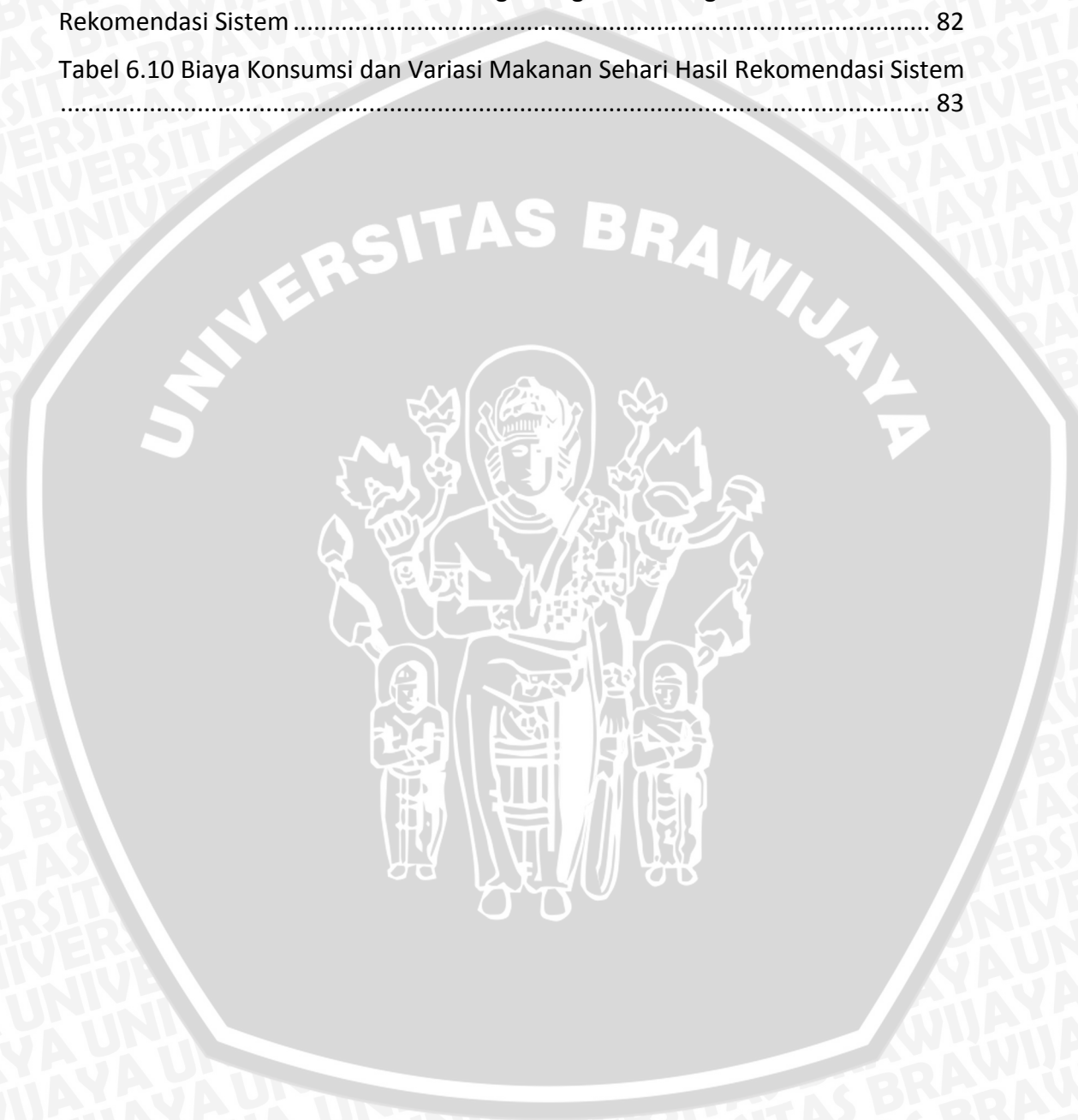
7.1 Kesimpulan.....	84
7.2 Saran	84
Daftar Pustaka	86
Lampiran A Tipe-tipe Anjuran Porsi Sehari	88
Lampiran B Daftar Bahan Makanan	89



DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka	6
Tabel 2.2 Perbandingan <i>Input</i> , <i>Proses</i> , dan <i>Output</i> Penelitian	7
Tabel 2. 3 Tabel Batas Ambang Indeks Massa Tubuh (IMT)	11
Tabel 2.4 Tabel Aktivitas Fisik	13
Tabel 4.1 Tabel Data Anggota Keluarga	38
Tabel 4. 2 Tabel Hasil Konversi Indeks Bahan Makanan	40
Tabel 4. 3 Tabel Hasil Konversi Nama Bahan Makanan	40
Tabel 4.4 Tabel Hasil Perhitungan Total Harga	41
Tabel 4. 5 Tabel Hasil Perhitungan Jumlah Berat Bahan Makanan	43
Tabel 4.6 Tabel Hasil Perhitungan Kandungan Kalori	43
Tabel 4.7 Tabel Hasil Perhitungan Kandungan Karbohidrat	44
Tabel 4.8 Tabel Hasil Perhitungan Kandungan Protein.....	45
Tabel 4.9 Tabel Hasil Perhitungan Kandungan Lemak.....	46
Tabel 4.10 Tabel Contoh Perubahan Kecepatan Partikel	49
Tabel 4.11Tabel Contoh Perubahan Posisi Partikel	50
Tabel 4.12 Data Bahan Makanan Hasil Perhitungan Manual	50
Tabel 4.13 Harga dan Berat Makanan Hasil Perhitungan Manual.....	51
Tabel 4.14 Kandungan Kalori Bahan Makanan Hasil Perhitungan Manual	52
Tabel 4.15 Kandungan Karbohidrat, Protein dan Lemak Bahan Makanan Hasil Perhitungan Manual.....	53
Tabel 4.16 Tabel Perancangan Uji Coba Berdasarkan Banyaknya Partikel.....	56
Tabel 4.17 Tabel Perancangan Uji Coba Berdasarkan Banyaknya Iterasi.....	56
Tabel 4.18 Tabel Perancangan Uji Coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}	57
Tabel 4.19 Tabel Perancangan Uji Coba Berdasarkan Batas Angka Permutasi	57
Tabel 6.1 Tabel Hasil Pengujian Jumlah Partikel.....	74
Tabel 6.2 Tabel Hasil Pengujian Kombinasi Nilai ω_{min} dan ω_{max}	76
Tabel 6.3 Tabel Hasil Pengujian Banyaknya Iterasi	77
Tabel 6. 4 Tabel Hasil Pengujian Batas Angka Permutasi	79
Tabel 6.5 Data Keluarga Berdasarkan Kuesioner	80

Tabel 6.6 Data Konsumsi Keluarga Beserta Biaya dalam Satu Hari.	81
Tabel 6.7 Kebutuhan Gizi Keluarga Aktual	81
Tabel 6.8 Kandungan Gizi Makanan Hasil Rekomendasi Sistem	81
Tabel 6.9 Selisih Kebutuhan Gizi Keluarga dengan Kandungan Gizi Makanan Hasil Rekomendasi Sistem	82
Tabel 6.10 Biaya Konsumsi dan Variasi Makanan Sehari Hasil Rekomendasi Sistem	83



DAFTAR GAMBAR

Gambar 2.1 Pseudo Code Algoritma <i>Particle swarm optimization</i>	19
Gambar 3. 1 Alur Tahapan Penelitian	20
Gambar 3. 2 Gambaran Umum Perancangan Sistem	22
Gambar 4.1 Proses Algoritma <i>Particle Swarm Optimization</i> dalam Optimasi Pemenuhan Gizi Secara Umum.....	25
Gambar 4.2 Proses Inisialisasi Partikel Awal.....	26
Gambar 4.3 Proses Penghitungan <i>Fitness</i>	27
Gambar 4.4 Proses Konversi Indeks Bahan Makanan.....	29
Gambar 4.5 Proses Hitung Berat Total	30
Gambar 4.6 Proses Hitung Harga Total.....	31
Gambar 4.7 Proses Perhitungan Kandungan Gizi	32
Gambar 4.8 Proses Perhitungan Variasi	33
Gambar 4.9 Proses Perhitungan Pbest	34
Gambar 4.10 Proses Perhitungan Gbest.....	35
Gambar 4.11 Proses Perhitungan <i>Velocity</i>	36
Gambar 4.12 Proses Perhitungan Posisi Partikel.....	37
Gambar 4.13 Inisialisasi Partikel Awal	39
Gambar 4.14 Rancangan Antarmuka Halaman Utama.....	54
Gambar 5.1 Implementasi Halaman Panel Input Parameter, Input dan Data Keluarga.....	71
Gambar 5.2 Implementasi Panel Kebutuhan Gizi	72
Gambar 5.3 Implementasi Panel Hasil Optimasi.....	72
Gambar 5.4 Implementasi Panel Keterangan Bahan Makanan.....	73
Gambar 6.1 Grafik Hasil Pengujian Jumlah Partikel.....	75
Gambar 6.2 Grafik Hasil Pengujian Kombinasi Nilai ω_{min} dan ω_{max}	77
Gambar 6.3 Grafik Hasil Pengujian Banyaknya Iterasi.....	78
Gambar 6. 4 Grafik Hasil Pengujian Batas Angka Permutasi	80

DAFTAR KODE PROGRAM

Kode Program 5.1 Inisialisasi Partikel Awal	58
Kode Program 5.2 Inisialisasi Kecepatan Awal.....	59
Kode Program 5.3 Perhitungan <i>Fitness</i>	60
Kode Program 5.4 Konversi Indeks Bahan Makanan	61
Kode Program 5.5 Konversi Nama Bahan Makanan	63
Kode Program 5.6 Perhitungan Berat Total Bahan Makanan.....	63
Kode Program 5.7 Perhitungan Selisih Kalori	64
Kode Program 5.8 Perhitungan Selisih Karbohidrat Total	65
Kode Program 5.9 Perhitungan Selisih Protein Total.....	66
Kode Program 5.10 Perhitungan Selisih Lemak Total	67
Kode Program 5.11 Perhitungan Harga Total	67
Kode Program 5.12 Perhitungan Variasi.....	68
Kode Program 5.13 Perhitungan <i>Pbest</i>	68
Kode Program 5.14 Perhitungan <i>Gbest</i>	69
Kode Program 5.15 Update Kecepatan Partikel	70
Kode Program 5.16 Update Posisi Partikel	70

BAB 1 PENDAHULUAN

1.1 Latar belakang

Permasalahan mengenai gizi di Indonesia maupun pada negara berkembang lainnya kebanyakan didominasi permasalahan kekurangan zat-zat gizi seperti protein, besi, yodium, dan vitamin, sedangkan yang terjadi di kota-kota besar atau maju didominasi oleh masalah kelebihan zat gizi atau obesitas (Supariasa, 2001). Menurut Peraturan Menteri Kesehatan tahun 2014 mengenai pedoman gizi seimbang, konsumsi pangan masyarakat Indonesia cenderung dibawah standar gizi seimbang. Berdasarkan Riset Kesehatan Dasar (Riskesdas) 2013, 93,5% konsumsi sayuran dan buah-buahan pada penduduk usia di atas 10 masih di bawah anjuran. Kualitas protein yang dikonsumsi rata-rata perorang perhari juga cenderung rendah karena sebagian besar berasal dari protein nabati seperti sereal dan kacang-kacangan. Untuk konsumsi makanan dan minuman berkadar gula tinggi, garam tinggi dan lemak tinggi, baik pada masyarakat perkotaan maupun perdesaan, masih cukup tinggi. Riskesdas 2007, 2010, 2013 menunjukkan bahwa presentase kasus kekurangan gizi di Indonesia masih cenderung tinggi. Kecenderungan prevalensi kurus (*wasting*) anak balita dari 13,6% menjadi 13,3% dan menurun 12,1%. Sedangkan kecenderungan prevalensi anak balita pendek (*stunting*) sebesar 36,8%, 35,6%, 37,2%. Prevalensi gizi kurang (*underweight*) berturut-turut 18,4%, 17,9% dan 19,6%. Prevalensi kurus anak sekolah sampai remaja berdasarkan Riskesdas 2010 sebesar 28,5%. (Permenkes, 2014).

Makanan sehari-hari yang dikombinasikan dengan baik, selain berfungsi sebagai asupan gizi, juga bermanfaat untuk banyak hal lainnya seperti menyuplai energi, mendukung proses pertumbuhan serta pemeliharaan jaringan tubuh serta membantu jalannya proses-proses yang terjadi dalam tubuh. Hal ini tentunya juga berpengaruh besar terhadap aktifitas fisik manusia sehari-hari. Bila asupan gizi tidak tercukupi maka kemungkinan besar aktifitas pun akan terganggu. Karena hal-hal di atas, kebutuhan asupan gizi adalah hal yang penting dalam kehidupan sehari-hari. Pendekatan medis dan pelayanan kesehatan saja tidak cukup untuk menanggulangi masalah gizi karena penyebab permasalahan gizi sendiri terdiri dari banyak faktor seperti pertanian, sosial, ekonomi dan budaya.

Dalam ilmu gizi, istilah Angka Kecukupan Gizi (AKG) sebagai acuan penentu kecukupan gizi. Angka Kecukupan Gizi dapat mengacu pada patokan berat badan, kelompok umur, gender, aktifitas fisik dan kondisi fisiologis tubuh tertentu seperti contohnya kehamilan dan menyusui. Kebutuhan gizi seseorang tentunya berbeda-beda berkaitan dengan faktor-faktor yang berpengaruh pada kebutuhan asupan gizi masing-masing individu, faktor-faktor seperti usia, berat badan, tinggi badan, jenis kelamin, dan aktivitas. Sebagai contoh seorang laki-laki dewasa dengan aktivitas yang memerlukan banyak tenaga tentunya memerlukan jumlah energi yang berbeda dengan anak kecil yang masih duduk di

bangku sekolah dasar. Maka dari itu dibutuhkan takaran bahan makanan yang tepat untuk memenuhi dengan kebutuhan gizi setiap orang.

Untuk mencukupi kebutuhan gizi, manusia harus mengonsumsi makanan sehat. Makanan sehat didefinisikan sebagai makanan yang memiliki perpaduan kandungan gizi yang seimbang untuk mencukupi kebutuhan gizi seseorang. Pola makan 4 sehat 5 sempurna dapat dikatakan sebagai makanan sehat karena mengandung semua zat gizi yang diperlukan tubuh (Almatsier, 2009). Dalam menentukan kombinasi bahan makanan apa saja yang harus diolah untuk memenuhi kebutuhan gizi sehari-hari tentu sulit, jika harus membandingkan kandungan gizi tiap bahan makanan dan kebutuhan gizi yang harus dipenuhi di tabel AKG. Dalam hal ini, aspek teknologi dapat membantu memberikan solusi yaitu rekomendasi kombinasi optimal bahan pangan yang dapat memenuhi kebutuhan gizi, tidak hanya satu orang tetapi kombinasi bahan makanan untuk memenuhi kebutuhan satu keluarga.

Untuk sistem rekomendasi optimasi kombinasi bahan makanan ini telah banyak penelitian terdahulu misalnya Rianawati & Mahmudy (2015) yang melakukan penelitian dengan menggunakan algoritma genetika untuk optimasi komposisi makanan bagi penderita diabetes mellitus, juga Pratiwi et al. (2014) yang melakukan serupa yaitu implementasi metode yang sama yaitu algoritma genetika untuk optimasi kebutuhan gizi. Sedangkan Afandi et al. (2014), melakukan penelitian dalam membangun sistem pendukung keputusan pemilihan menu makanan sehat dan bergizi menggunakan metode K-Nearest Neighbor.

Pada penelitian ini dilakukan optimasi pemenuhan kebutuhan gizi keluarga dengan rekomendasi kombinasi bahan makanan. Untuk proses optimasi pada penelitian ini, algoritma yang digunakan adalah algoritma *particle swarm optimization* (PSO). Kennedy dan Eberhard pertama kali memperkenalkan algoritma *particle swarm optimization* (PSO) pada tahun 1995. Secara umum, algoritma *Particle swarm optimization* (PSO) merupakan salah satu teknik optimasi berbasis populasi untuk mencari solusi optimal menggunakan populasi dari partikel itu sendiri. Ide dasar dari PSO adalah bahwa setiap kerumunan partikel merupakan solusi dari ruang solusi (Kurniawan, 2010). Kelebihan dari algoritma PSO sendiri yaitu, algoritma ini memiliki konsep sederhana, dapat diimplementasikan dengan mudah, dan dibandingkan dengan algoritma matematis dan teknis optimasi heuristik lainnya, algoritma PSO dinilai lebih efisien dalam hal perhitungan (Maickel, et al., 2009). Dari beberapa hal yang telah dielaskan di atas, maka penulis memutuskan untuk melakukan penelitian yang berjudul "Implementasi Algoritma *Particle Swarm Optimization* dalam Optimasi Pemenuhan Kebutuhan Gizi Keluarga".

1.2 Rumusan masalah

Dari latar belakang yang telah dijabarkan pada bagian sebelumnya, berikut akan dirumuskan beberapa permasalahan yang akan dibahas yaitu:

1. Bagaimana menerapkan algoritma *particle swarm optimization* dalam penyelesaian permasalahan Optimasi Pemenuhan Kebutuhan Gizi Keluarga?
2. Bagaimana hasil rekomendasi yang didapat dari hasil optimasi pemenuhan kebutuhan gizi keluarga menggunakan algoritma *particle swarm optimization*?

1.3 Tujuan

Berdasarkan dengan permasalahan yang sudah dirumuskan di atas, tujuan dari penyusunan tugas akhir ini adalah:

1. Menerapkan parameter algoritma *particle swarm optimization* dalam penyelesaian permasalahan Optimasi Pemenuhan Kebutuhan Gizi Keluarga.
2. Mengetahui hasil rekomendasi yang didapat dari hasil optimasi menggunakan algoritma *particle swarm optimization*.

1.4 Manfaat

Dari pembuaatan tugas akhir ini diharapkan menghasilkan manfaat bagi banyak pihak diantaranya, masyarakat luas untuk membantu memilih bahan pangan yang sesuai untuk memenuhi kebutuhan gizi keluarga dan juga untuk penulis yaitu mampu menerapkan algoritma *particle swarm optimization* dalam optimasi pemenuhan kebutuhan gizi keluarga dan mampu mengukur kualitas rekomendasi yang didapat dari hasil optimasi menggunakan algoritma *particle swarm optimization* juga sebagai syarat kelulusan penulis dan mencapai gelar sarjana.

1.5 Batasan masalah

Berdasarkan rumusan masalah dan tujuan yang telah dijabarkan, berikut adalah beberapa batasan masalah dalam penulisan tugas akhir ini:

1. Penelitian dikhususkan hanya untuk keluarga sehat yang tidak menderita penyakit apapun atau mempunyai riwayat penyakit turunan.
2. Asupan gizi yang direkomendasikan dibatasi pada rekomendasi persentase karbohidrat, protein, lemak dan energi
3. Hasil *output* yang diharapkan adalah rekomendasi bahan makanan yang bisa dikonsumsi satu keluarga untuk memenuhi kebutuhan gizi dalam jangka waktu satu minggu.

1.6 Sistematika pembahasan

Sistematika penulisan dalam skripsi ini sebagai berikut:

- | | |
|----------------|--|
| BAB I | Pendahuluan
Memuat latar belakang penulisan tugas akhir, rumusan masalah, batasan masalah, tujuan, manfaat dari implementasi algoritma <i>particle swarm optimization</i> untuk optimasi pemenuhan kebutuhan gizi keluarga. |
| BAB II | Landasan Kepustakaan
Kajian pustaka membahas penelitian yang telah dilakukan dan yang diusulkan. Dasar teori memaparkan teori dasar dan teori pendukung yang berhubungan dengan metode-metode yang digunakan dalam implementasi algoritma <i>particle swarm optimization</i> untuk optimasi pemenuhan kebutuhan gizi keluarga. |
| BAB III | Metode Penelitian
Membahas metode penelitian yang digunakan dalam implementasi algoritma <i>particle swarm optimization</i> untuk optimasi pemenuhan kebutuhan gizi keluarga. |
| BAB IV | Perancangan
Membahas tentang tahap perancangan dari sistem yaitu dalam implementasi algoritma <i>particle swarm optimization</i> untuk optimasi pemenuhan kebutuhan gizi keluarga. |
| BAB V | Implementasi
Membahas tentang tahap implementasi berdasarkan tahap perancangan dari sistem yaitu dalam implementasi algoritma <i>particle swarm optimization</i> untuk optimasi pemenuhan kebutuhan gizi keluarga. |
| BAB VI | Pengujian dan Analisis
Mencakup tentang hasil pengujian dan juga analisis sistem terhadap sistem yang telah direalisasikan. |
| BAB VII | Penutup
Memuat kesimpulan yang diperoleh dari hasil akhir dalam Implementasi Algoritma <i>Particle swarm optimization</i> untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga. |

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini memuat tinjauan pustaka meliputi kajian pustaka dan dasar teori yang dinilai perlu dalam penelitian ini. Kajian pustaka akan membahas penelitian sebelumnya dan yang diusulkan dan yang telah dilakukan, sedangkan Dasar teori membahas teori yang mendukung dengan penyusunan penelitian yang diusulkan.

2.1 Kajian Pustaka

Kajian pustaka akan berisi tentang pembahasan penelitian terdahulu dan juga yang diusulkan. Kajian pustaka pada penelitian ini diambil dari penelitian-penelitian relevan sebelumnya yang pernah dilakukan. Penelitian terdahulu yang juga relevan dengan permasalahan yang sedang dibahas dalam penelitian ini adalah salah satu *data* pendukung yang dirasa perlu oleh peneliti untuk dijadikan bagian tersendiri. Dasar atau acuan yang berupa teori-teori atau temuan-temuan melalui hasil berbagai penelitian sebelumnya adalah hal penting yang dapat dijadikan sebagai *data* pendukung.

Untuk sistem rekomendasi optimasi kombinasi bahan makanan ini telah banyak penelitian terdahulu seperti halnya Rianawati & Mahmudy (2015) yang melakukan penelitian mengenai optimasi komposisi makanan bagi penderita diabetes mellitus dengan menggunakan algoritma genetika. Representasi kromosom yang digunakan pada penelitian ini adalah representasi permutasi dengan bilangan *integer* yang berisi nomor makanan yang akan dikonsumsi, lalu untuk metode *crossover* yang digunakan yaitu *single-point crossover*, dan metode mutasi yang digunakan yaitu *reciprocal exchange mutation*. Nilai *fitness* didapat dari pembagian antara pembagian konstanta C dengan penalti gizi ditambah dengan total harga. Hasil akhir dari penelitian ini berupa rekomendasi bahan makanan sejumlah panjang kromosom yang terpilih beserta kandungan gizi dan harganya.

Pada kajian pustaka lainnya, Pratiwi et al. (2014) melakukan penelitian serupa yaitu implementasi metode yang sama algoritma genetika untuk optimasi kebutuhan gizi. Pada penelitian tersebut, representasi kromosom yang digunakan adalah bilangan *random* antara 1 hingga 125 yang merupakan nomor indeks bahan makanan. agar hasil *offspring* lebih beragam panjang kromosom divariasikan berkisar 5 sampai 15. Proses *crossover* yang digunakan adalah metode *one cut point crossover* dan proses mutasi dilakukan dengan menggantikan sejumlah gen dalam kromosom dengan bilangan acak antara 1 sampai 125. Sementara seleksi *elitism* digunakan untuk metode seleksi. Nilai *fitness* diperoleh dari pengurangan antara total harga 10 bahan makanan tertinggi dengan jumlah harga pada kromosom dan penalti gizi. Hasil akhir dari penelitian ini adalah rekomendasi bahan makanan sejumlah panjang kromosom yang terpilih beserta kandungan gizi dan harganya.

Penulis melakukan penelitian mengenai penggunaan algoritma *Particle swarm optimization* untuk optimasi pemenuhan kebutuhan gizi keluarga. Hasil dari penelitian ini berupa susunan bahan makanan selama jumlah hari yang diinginkan beserta takaran untuk masing-masing anggota keluarga dan juga harga. Penggunaan algoritma ini diharapkan dapat mengoptimasi pemenuhan kebutuhan gizi keluarga dengan baik dan juga dapat menghasilkan dengan biaya minimum dan tingkat variasi bahan makanan yang banyak. Perbandingan penelitian sebelumnya dengan penelitian yang akan dilakukan ditunjukkan pada Tabel 2.1.

Tabel 2.1 Tabel Kajian Pustaka

NO	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Tugas Akhir Penulis
1	Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi	Monica Intan Pratiwi, Wayan Fidaus Mahmudy, Candra Dewi	Menggunakan Algoritma Genetika dan rekomendasi hanya untuk satu orang	Menggunakan Algoritma <i>Particle swarm optimization</i> dan rekomendasi untuk satu keluarga
2	Implementasi Algoritma Genetika Untuk Optimasi Komposisi Makanan Bagi Penderita Diabetes Mellitus	Artika Rianawati, Wayan Firdaus Mahmudy	Menggunakan Algoritma Genetika dan rekomendasi hanya untuk satu orang yang menderita penyakit diabetes mellitus	Menggunakan Algoritma <i>Particle swarm optimization</i> dan rekomendasi untuk satu keluarga yang tidak menderita penyakit apapun
3	<i>A Novel Bankruptcy Prediction Model Based on An Adaptive Fuzzy K-Nearest Neighbor Method</i>	Hui-Ling Chen, Bo Yang, Gang Wang, Jie Liu, Xin Xu, Su-Jing Wang, Da-You Liu	Menggunakan model usulan FKNN-PTVPSO untuk penyelesaian masalah prediksi kebangkrutan perusahaan	Menggunakan algoritma <i>Particle swarm optimization</i> untuk menghasilkan rekomendasi bahan makanan untuk memenuhi kebutuhan gizi keluarga

4	Implementasi metode <i>k-nearest neighbor</i> untuk pendukung keputusan pemilihan menu makanan sehat dan bergizi	Mochamad Noor Afandie	Menggunakan teknik Sistem pendukung keputusan dengan metode <i>k-nearest neighbor</i>	Menggunakan Teknik optimasi dengan algoritma <i>Particle swarm optimization</i>
5	Implementasi Algoritma <i>Particle swarm optimization</i> Untuk Optimasi Fungsi Keanggotaan Sistem Inferensi Fuzzy Mamdani pada Penyakit Hepatitis	Ratna Putri Prati Sasya, Candra Dewi, S, Indriati	Permasalahan yang diangkat adalah optimasi fungsi keanggotaan sistem inferensi fuzzy mamdani pada penyakit hepatitis	Permasalahan yang diangkat tentang optimasi kebutuhan gizi

Pada Tabel 2.2 dibawah ini ditunjukkan perbandingan *input*, proses dan juga *output* dari penelitian sebelumnya dengan penelitian yang dilakukan penulis.

Tabel 2.2 Perbandingan *Input*, Proses, dan *Output* Penelitian

No	Judul	Input	Proses	Output
1	Implementasi Algoritma Genetika pada Optimasi Biaya Pemenuhan Gizi (Pratiwi, Mahmudy, & Dewi, 2014)	-Usia - Aktivitas - Tinggi badan (cm) - Berat badan (kg) - Jenis kelamin	Inisialisasi populasi awal - Perhitungan nilai <i>fitness</i> - <i>Crossover</i> - Mutasi - Seleksi <i>elitism</i> - Proses berhenti pada iterasi tertentu	Rekomendasi bahan-bahan makanan yang mencukupi kebutuhan gizi selama satu hari beserta kandungan gizi dengan harga minimum

2	Implementasi Algoritma Genetika Untuk Optimasi Komposisi Makanan Bagi Penderita Diabetes Mellitus Rianawaty & Mahmudy, 2015	-Usia - Jenis kelamin - Aktivitas - Berat badan (kg) - Tinggi badan (cm) - Prioritas lemak - Prioritas Protein - Prioritas Karbohidrat	-Inisialisasi populasi awal - Perhitungan nilai <i>fitness</i> - <i>Crossover</i> - Mutasi - Seleksi <i>elitism</i> - Proses berhenti pada iterasi tertentu	Rekomendasi bahan makanan untuk penderita diabetes melitus yang mencukupi kebutuhan gizi selama satu hari beserta kandungan gizinya beserta harga minimum
3	<i>A Novel Bankruptcy Prediction Model Based on An Adaptive Fuzzy K-Nearest Neighbor Method</i> (Hui-Ling Chen, Bo Yang, Gang Wang, Jie Liu, Xin Xu, Su-Jing Wang, Da-You Liu, 2011)	- Kelas - Ketetangaan - Jarak ketetangaan - Jumlah populasi	- Inisialisasi partikel - Inisialisasi populasi - Update posisi dan kecepatan - Pelatihan FKNN - Perhitungan <i>fitness</i> - Mencari pfit dan pbest - Mencari gfit dan gbest	Prediksi kebangkrutan perusahaan dan hasil perbandingan metode yaitu Model usulan FKNN-PTVPSO menghasilkan hasil klasifikasi yang lebih baik dibanding metode KNN saja
4	Implementasi metode k-nearest neighbor untuk pendukung keputusan pemilihan menu makanan sehat (Affandie, 2013)	-Usia - Berat badan (kg) - Tinggi badan (cm) - Jenis kelamin - Aktivitas	- data <i>pre processing</i> (menghitung kebutuhan gizi) -klasifikasi metode KNN	Rekomendasi menu makanan sehat untuk memenuhi kebutuhan gizi yang sesuai dengan kondisi seseorang

5	Implementasi Algoritma <i>Particle swarm optimization</i> Untuk Optimasi Fungsi Keanggotaan Sistem Inferensi Fuzzy Mamdani pada Penyakit Hepatitis	- parameter algoritma $\omega_{min}, \omega_{max}, C_1, C_2$, jumlah partikel, iterasi maksimal)	- proses fuzzifikasi untuk mendapatkan variabel linguistik himpunan fuzzy. yang kemudian akan dioptimasi oleh algoritma <i>Particle swarm optimization</i> . - inialisasi partikel awal - inialisasi kecepatan partikel awal - perhitungan nilai cost semua partikel dengan <i>Mean Square Error</i> (MSE) - menentukan pbest dan gbest - menghitung kecepatan partikel - menghitung posisi partikel - berhenti pada jumlah iterasi maksimal	Hasil akhir dari sistem ini adalah batas linguistik baru.
---	--	---	---	---

2.2 Makanan Sehat dan Gizi

Makanan sehat didefinisikan sebagai makanan yang memiliki perpaduan kandungan gizi yang seimbang untuk mencukupi kebutuhan gizi seseorang/ Pola makan 4 sehat 5 sempurna dapat dikatakan sebagai makanan sehat karena mengandung semua zat gizi yang diperlukan tubuh (Almatsier, 2010).

Pola 4 sehat 5 sempurna pada umumnya terdiri dari kombinasi makanan berikut:

1. Makanan pokok guna memberikan rasa kenyang, seperti nasi, jagung, ubi, singkong atau hasil olahan tepung seperti mie dan bahan lainnya.
2. Lauk pauk yang terdiri dari hewani dan nabati. Lauk hewani seperti daging, ayam, ukan, kerang dan telur sedangkan lauk nabati seperti kacang kedelai, kacang hijau, tahu dan tempe

3. Sayur mayur seperti daun-daunan, umbi-umbian, kacang-kacangan, dan sebagainya.
4. Buah sebagai pencuci mulut seperti pepaya, pisang, jeruk, dan sebagainya
5. Susu atau bahan olahan susu yang diajurkan untuk golongan yang membutuhkan asupan protein lebih banyak seperti balita, ibu hamil dan ibu menyusui.

Gizi atau nutrisi adalah substansi organik yang diperlukan untuk aktivitas metabolisme di dalam tubuh. Gizi didapatkan dari makanan yang dikonsumsi tubuh yang melalui proses asimilasi. Tidak semua nutrisi yang terkandung pada makanan diperlukan tubuh. Nutrisi yang tidak diperlukan tubuh akan dikeluarkan kembali untuk mempertahankan kehidupan dan fungsi normal organ-organ dalam tubuh. Besar nutrisi yang diperlukan setiap orang berbeda-beda, apabila seseorang kekurangan atau kelebihan gizi akan berpengaruh pada kesehatan tubuh seseorang tersebut (Almatsier, 2010)

Masalah gizi di Indonesia maupun di negara berkembang lainnya masih didominasi oleh masalah kekurangan zat-zat gizi seperti protein, besi, yodium dan vitamin, sedangkan yang terjadi di kota-kota besar Negara berkembang didominasi oleh masalah kelebihan zat gizi atau obesitas (Supariasa, 2001). Padahal, makanan sehari-hari yang dikombinasikan dengan baik berfungsi sebagai asupan gizi yang bermanfaat banyak seperti memberi energi, membantu pertumbuhan dan pemeliharaan jaringan tubuh serta membantu jalannya proses-proses yang terjadi dalam tubuh. Hal ini tentunya juga berpengaruh besar terhadap aktifitas fisik manusia sehari-hari. Bila asupan gizi tidak tercukupi maka kemungkinan besar aktifitas pun akan terganggu.

Kebutuhan asupan gizi adalah hal yang penting dalam kehidupan sehari-hari, pendekatan medis dan pelayanan kesehatan saja tidak cukup untuk menanggulangi masalah gizi karena penyebab permasalahan gizi sendiri terdiri dari banyak faktor seperti pertanian, sosial, ekonomi dan budaya.

2.3 Angka Kecukupan Gizi (AKG)

Angka Kecukupan Gizi (AKG) adalah tingkat konsumsi zat gizi esensial yang ditinjau cukup untuk memenuhi kebutuhan gizi mayoritas orang sehat. Berbeda dengan Angka Kebutuhan Gizi yang diartikan sebagai jumlah minimal zat-zat gizi yang diperlukan oleh seseorang untuk mempertahankan status gizi yang memadai. Angka Kecukupan Gizi dapat di dasarkan dengan patokan berat badan, kelompok umur, gender, aktifitas fisik dan kondisi fisiologis tubuh tertentu seperti contohnya kehamilan dan menyusui (Almatsier, 2009).

Kegunaan AKG antara lain (Afandie, et al., 2014):

1. Sebagai acuan untuk menilai kecukupan gizi
2. Sebagai acuan untuk menyusun makanan sehari-hari termasuk perencanaan makanan di institusi.

3. Sebagai dasar perhitungan dalam perencanaan penyediaan pangan tingkat regional maupun nasional.
4. Sebagai dasar pendidikan gizi
5. Sebagai dasar label pangan mencantumkan nilai informasi gizi

2.4 Indeks Massa Tubuh (IMT)

Indeks Massa Tubuh (IMT) adalah cara sederhana untuk memantau status gizi untuk kekurangan dan kelebihan konsumsi gizi dan dalam rangka mencapai angka harapan hidup lebih panjang. IMT digunakan untuk menyatakan batasan berat badan orang dewasa normal sejak tahun 1895 oleh WHO dengan nama *Body Mass Indeks* (BMI) (Supariasa, 2001). Berat badan seseorang dinyatakan normal, kurus atau gemuk dapat diketahui dengan IMT. Penggunaan IMT dibatasi hanya untuk orang dewasa berumur > 18 tahun dan tidak dapat diterapkan pada bayi, anak, remaja, ibu hamil, dan olahragawan. (Depkes, 2014)

Rumus perhitungan IMT:

$$IMT = \frac{\text{Berat badan (Kg)}}{\text{Tinggi Badan(m)} \times \text{Tinggi Badan(m)}} \quad (2-1)$$

Hasil perhitungan IMT selanjutnya dibandingkan dengan tabel ambang batas yang ditentukan oleh Departemen Kesehatan Indonesia. Batas ambang IMT atau perbedaan antara batas ambang untuk laki-laki dan perempuan ditentukan berdasarkan ketentuan FAO/WHO. Batas ambang normal yang disebutkan adalah, untuk laki-laki adalah: 20,1–25,0; dan untuk perempuan adalah: 18,7–23,8. Untuk kepentingan pemantauan dan tingkat defisiensi kalori ataupun tingkat kegemukan, lebih lanjut FAO/WHO menyarankan menggunakan satu batas ambang antara laki-laki dan perempuan. Ketentuan yang digunakan adalah menggunakan ambang batas laki-laki untuk kategori kurus tingkat berat dan menggunakan ambang batas pada perempuan untuk kategori gemuk tingkat berat. Berdasarkan pengalaman klinis dan hasil penelitian di beberapa negara berkembang Untuk kepentingan Indonesia, batas ambang dimodifikasi lebih lanjut. Pada akhirnya diambil kesimpulan, batas ambang IMT untuk Indonesia ditunjukkan pada Tabel 2.2 berikut:

Tabel 2. 3 Tabel Batas Ambang Indeks Massa Tubuh (IMT)

Kategori	Keterangan	IMT
Kurus	Kekurangan berat badan tingkat berat	< 17,0
Kurus	Kekurangan berat badan tingkat ringan	<17,5 – 18,5
Normal	-	18,5 – 25,0
Gemuk	Kelebihan berat badan tingkat ringan	25,0 – 27,0
Gemuk	Kelebihan berat badan tingkat berat	27,0

2.5 Angka Metabolisme Basal (AMB)

Angka metabolisme tubuh (AMB) adalah kebutuhan energi minimal yang diperlukan tubuh untuk melakukan proses dalam tubuh. AMB dinyatakan dalam kilokalori per kilogram berat badan per jam. Faktor-faktor yang mempengaruhi AMB adalah sebagai berikut (Almatsier, 2010):

a. Ukuran tubuh

Perbedaan ukuran tubuh menyebabkan perbedaan hasil AMB. Setiap perbedaan 10 kg mempunyai perbedaan AMB sekitar 120 kkal per hari.

b. Jenis kelamin

Perempuan mempunyai lebih banyak jaringan lemak dan lebih sedikit otot daripada laki-laki sehingga AMB perempuan lebih rendah 5% dari laki-laki.

c. Umur

AMB tinggi dimiliki bayi baru lahir dan meningkat hingga usia 2 tahun, kemudian menurun hingga mengalami peningkatan saat usia remaja. AMB turun setiap 10 tahun dengan prosentasi penurunan sebesar 2% setelah umur 30 tahun.

d. Tidur

Saat tidur, AMB mengalami penurunan sekitar 10%.

e. Suhu tubuh

Setiap kenaikan suhu tubuh sebesar 1°C maka terjadi kenaikan IMB sebesar 13%.

f. Status gizi

Keadaan gizi kurang menurunkan AMB sebesar 20%.

Cara menentukan AMB dengan menggunakan Rumus Harris Benedict (199) dijelaskan dengan Persamaan 2.3 dan Persamaan 2.4 berikut ini (Instalasi Gizi Perjan RS Dr. Cipto Mangunkusumo dan Asosiasi Dietisien Indonesia, 2010):

- Laki-laki
$$IMB = 66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U) \quad (2.3)$$

- Perempuan
$$IMB = 655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U) \quad (2.4)$$

Keterangan:

IMB = Indeks Massa Basal

BB = Berat Badan

TB = Tinggi Badan

U = Umur

2.6 Aktivitas Fisik

Aktivitas fisik atau *Physical Activity* (PA) mempengaruhi kebutuhan energi yang dibutuhkan sehari-hari karena menggunakan otot sebagai penunjang gerakannya. Lama dan tingkat berat aktivitas juga mempengaruhi kebutuhan energi yang diperlukan seseorang per hari (Almatsier, 2010).

Menurut Departemen Kesehatan RI, aktivitas fisik dikelompokkan menjadi 3 yaitu ringan, sedang dan berat. Sedangkan faktor aktivitas antara laki-

laki dan perempuan juga berbeda. Berikut merupakan penjelasan dari kelompok aktivitas fisik menurut Depkes RI ditunjukkan dengan Tabel 2.2

Tabel 2.4 Tabel Aktivitas Fisik

[Sumber: Depkes RI]

Kelompok Aktivitas	Jenis Kegiatan	Jenis Kelamin	Faktor Aktivitas
Ringan	75% dari waktu untuk berdiri atau duduk.	Laki-laki	1,58
	25% untuk kegiatan berdiri dan berpindah	Perempuan	1,45
Sedang	25% waktu untuk duduk atau berdiri.	Laki-laki	1,67
	75% waktu untuk kegiatan kerja	Perempuan	1,55
Berat	40% waktu untuk duduk dan berdiri.	Laki-laki	1,88
	60% waktu untuk kegiatan kerja	Perempuan	1,75

2.7 Zat Gizi

Untuk menjalankan fungsinya, zat gizi merupakan senyawa yang dibutuhkan oleh tubuh. Fungsi dari zat gizi antara lain adalah penghasil energi, pembangun, dan pemelihara jaringan, dan pengatur proses-proses yang ada dalam tubuh. Zat-zat gizi yang dibutuhkan tubuh meliputi karbohidrat, lemak, protein, energi, mineral, dan vitamin. Zat-zat tersebut banyak ditemukan pada berbagai macam bahan makanan seperti daging, ikan, sayur, buah, dan lain-lain (Almatsier, 2009). Karena pentingnya peran zat gizi bagi tubuh, semua makanan yang masuk ke dalam tubuh harus diperhatikan dengan baik kandungan gizinya. Kekurangan konsumsi zat gizi menyebabkan berbagai penyakit seperti busung lapar, sedangkan kelebihan gizi menyebabkan kegemukan atau obesitas.

2.7.1 Karbohidrat

Karbohidrat berperan penting sebagai sumber energi utama bagi makhluk hidup, lebih tepatnya bagi manusia dan hewan. Semua sumber karbohidrat berasal dari tumbuhan yang melalui proses fotosintesis, klorofil dari tanaman dapat membentuk karbohidrat dari karbon dioksida (CO_2) yang berasal dari udara dan air (H_2O) yang berasal dari tanah. Karbohidrat yang dihasilkan adalah karbohidrat sederhana glukosa. Di negara-negara berkembang sekitar 80% energi berasal dari makanan yang mengandung karbohidrat. Nilai energi dari karbohidrat adalah 4 kkal per gram. (Almatsier, 2009).

Karbohidrat diklasifikasikan dalam dua golongan yaitu karbohidrat sederhana dan karbohidrat kompleks. Dalam satu molekulnya karbohidrat kompleks memiliki lebih dari dua unit gula sederhana.

Karbohidrat sederhana terdiri atas:

1. Monosakarida
2. Disakarida
3. Gula alkohol
4. Oligosakarida

Karbohidrat kompleks terdiri atas:

1. Polisakarida
2. Serat / Polisakarida non pati

2.7.2 Lemak

Lemak adalah nutrisi kedua yang digunakan tubuh sebagai bahan bakar untuk menghasilkan energi. (Hartono, 2006). Kategori lemak terbagi menjadi dua, yaitu berdasarkan sumbernya dan berdasarkan wujud. Lemak berdasarkan sumber terdiri dari lemak nabati (lemak yang terkandung dalam tumbuh-tumbuhan seperti kacang-kacangan) dan lemak hewani (lemak yang berasal dari hewan yang mengandung vitamin A dan D). Sedangkan berdasarkan wujudnya terbagi menjadi lemak padat dan cair.

Minyak tumbuh-tumbuhan (minyak kelapa, kelapa sawit, kacang tanah, kacang kedelai, jagung), mentega, margarin, daging, dan ayam adalah sumber utama lemak. Sumber lemak lain antara lain kacang-kacangan, biji-bijian, krim, susu, keju, kuning telur, dan masakan yang menggunakan minyak.

Kekurangan konsumsi lemak menyebabkan gangguan pada syaraf dan penglihatan, menghambat pertumbuhan bayi dan anak-anak, gagalnya proses reproduksi, serta gangguan pada organ tubuh seperti hati dan ginjal. Kelebihan konsumsi lemak dapat menyebabkan obesitas serta peningkatan kolesterol di dalam tubuh yang memicu timbulnya penyakit berbahaya

2.7.3 Protein

Protein memegang peranan penting dalam melaksanakan fungsi fisiologi yang penting. Protein terbentuk dari berbagai asam amino yang terangkai dalam ikatan peptida. Berdasarkan fungsinya, protein dibagi dari tiga kelompok yaitu protein lengkap, protein setengah lengkap dan protein tidak lengkap.

Protein lengkap berfungsi untuk pertumbuhan, penggantian jaringan tubuh yang rusak juga untuk keperluan lain seperti pembentukan enzim, hormon, antibodi serta energi jika perlu. Contoh bahan makanan yang merupakan protein lengkap adalah telur dan susu, kedua bahan makanan ini mengandung seluruh asam amino esensial yang mencukupi kebutuhan bagi pertumbuhan.

Protein setengah juga memiliki fungsi yang sama seperti protein lengkap kecuali fungsi pertumbuhan karena tidak memiliki kandungan asam amino yang

cukup untuk pembentukan jaringan tubuh yang baru. Contoh dari protein setengah lengkap adalah sumber protein hewani selain telur dan susu, seperti daging, ikan dan ayam.

Protein tidak lengkap pada umumnya bersumber dari bahan makanan nabati seperti kacang-kacangan dan sayur sayuran tidak dapat digunakan untuk pertumbuhan dan perbaikan jaringan yang rusak. Maka dari itu harus dikombinasikan dengan bahan makanan yang memiliki kandungan protein lebih lengkap. (Hartono, 2006)

Protein memiliki delapan kategori fungsi yaitu :

1. Membangun jaringan tubuh yang baru
2. Memperbaiki jaringan tubuh
3. Menghasilkan senyawa esensial
4. Mengatur tekanan osmotik
5. Mengatur keseimbangan cairan elektrolit dan asam basa
6. Menghasilkan mekanisme transportasi
7. Menghasilkan pertahanan tubuh
8. Menghasilkan energi

Satu gram protein menghasilkan energi sebanyak 4 kkal atau 16 kJ bila dioksidasi sebagai bahan bakar.

2.7.4 Energi

Energi atau kalori merupakan kemampuan seseorang untuk melakukan suatu aktivitas fisik sehari-hari. Energi dapat diperoleh dari konsumsi makanan yang mengandung karbohidrat, lemak, dan protein. Komposisi makanan mempunyai kandungan kalori yang berbeda-beda, tergantung jenis dan berat dari makanan itu sendiri. Energi sangat penting bagi tubuh karena sel dalam tubuh selalu melakukan aktivitas secara terus-menerus seperti membuat senyawa, menjalankan kerja mekanik dalam pergerakan, melakukan transportasi senyawa ke dalam seluruh tubuh, dan menghasilkan panas

Kebutuhan energi menurut WHO adalah konsumsi energi yang berasal dari makanan yang diperlukan seseorang untuk melakukan aktivitas dilihat dari jumlah energi yang dikeluarkan tubuh, ukuran tubuh, serta jenis aktivitas yang dilakukan. Kebutuhan energi pada anak-anak digunakan untuk pembentukan jaringan, ibu hamil dan menyusui digunakan untuk sekresi ASI. Sedangkan untuk orang dewasa, kebutuhan energi digunakan untuk metabolisme basal, aktivitas fisik, dan pengaruh dinamik khusus (SDA). Komponen SDA merupakan komponen sampingan sehingga dapat diabaikan (Almatsier, 2009). Ada dua hal yang diperlukan dalam perhitungan energi total yang diperlukan, yaitu Angka Metabolisme Basal (AMB) dan aktifitas fisik.

2.8 Optimasi

Optimasi adalah usaha untuk memperoleh kondisi terbaik untuk suatu masalah (Dorigo, M., dan Stutzle, T., 2004). Bidang rekayasa atau *engineering*

merupakan bidang ilmu yang senantiasa dihadapkan pada masalah optimasi dalam melakukan perancangan maupun dalam penyelesaian masalah.

Masalah yang dihadapi biasanya dinyatakan dalam bentuk suatu fungsi objektif atau fungsi biaya (*cost function*) yang nilainya dipengaruhi oleh beberapa parameter atau variabel. Masalah optimasi dikaitkan dengan batasan, yaitu kondisi yang harus dipenuhi oleh semua variabel.

2.9 Algoritma *Particle swarm optimization* (PSO)

Dr. Eberhart dan Dr. Kennedy pada tahun 1995 pertama kali memperkenalkan Algoritma PSO. Kehidupan populasi burung dan ikan dalam bertahan hidup menjadi acuan dalam algoritma optimasi ini (Haupt, 2004). *Particle swarm optimization* (PSO) merupakan salah satu teknik komputasi evolusioner, yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang *random* yang disebut dengan partikel. (Maickel, et al., 2009)

Pada dasarnya algoritma ini merupakan teknik optimasi berbasis populasi untuk mencari solusi optimal dengan menggunakan populasi dari partikel itu sendiri. PSO didasari ide bahwa setiap kerumunan partikel merupakan solusi dari ruang solusi (Kurniawan, 2010). Algoritma PSO seringkali digunakan untuk berbagai permasalahan optimasi baik kontinyu maupun diskrit, linier maupun nonlinier. PSO memodelkan aktivitas pencarian solusi terbaik dalam suatu ruang solusi sebagai aktivitas terbangnya kelompok partikel dalam suatu ruang solusi tersebut. Posisi partikel dalam ruang solusi tersebut merupakan kandidat solusi yang berisi variabel-variabel optimasi. Setiap posisi tersebut akan dikaitkan dengan sebuah nilai yang disebut nilai objektif atau nilai *fitness* yang dihitung berdasarkan fungsi objektif dari masalah optimasi yang akan diselesaikan.

Pada analogi burung, fungsi tersebut dapat berupa kualitas maupun kuantitas makanan pada tiap tempat dan kumpulan partikel atau dalam hal ini burung, akan mencari tempat dengan kualitas terbaik dan kuantitas terbanyak. Tidak seperti halnya metode lain, dalam fungsi kontinyu optimasi, PSO tidak menggunakan gradien informasi dalam pencarian solusi, sehingga tidak berakibat kesalahan fungsi persyaratan terus menerus (Nouaouria, 2013).

Beberapa istilah yang umumnya terdapat padai *particle swarm optimization* adalah berikut berikut :

1. *Swarm* : populasi dalam suatu algoritma.
2. Partikel: anggota (partikel) dalam suatu *swarm*. Setiap partikel merepresentasikan sebuah solusi yang potensial untuk permasalahan yang diselesaikan. Posisi dari suatu partikel adalah ditentukan oleh representasi solusi saat itu.
3. *Pbest* (*Personal best*): posisi *Pbest* suatu partikel yang menunjukkan posisi partikel yang dipersiapkan untuk mendapatkan solusi terbaik.
4. *Gbest* (*Global best*): posisi terbaik partikel pada *swarm*.
5. *Velocity* (vektor): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu partikel diperlukan untuk berpindah untuk memperbaiki posisinya semula.

6. *Inertia weight*: inertia weight dengan simbol ω , parameter ini digunakan sebagai pengontrol dampak dari adanya *velocity* yang diberikan oleh suatu partikel.
7. Koefisien akselerasi: koefisien akselerasi mempengaruhi jarak maksimum yang dapat diambil oleh sebuah partikel dalam sebuah iterasi. Biasanya bernilai 2 untuk kedua parameter C_1 dan C_2 (Sedighizadeh, et al., 2009).

Prosedur standar untuk menerapkan algoritma PSO adalah sebagai berikut (Maickel, et al., 2009):

1. Menginisialisasi populasi dari partikel dengan posisi dan *velocity* secara *random* dalam suatu ruang dimensi penelusuran.
2. Mengevaluasi fungsi *fitness* optimisasi yang diinginkan di dalam variabel d pada setiap partikel.
3. Membandingkan evaluasi *fitness* partikel dengan $Pbest$ -nya. Jika nilai yang ada lebih baik dibandingkan dengan nilai $Pbest$ -nya, maka nilai tersebut diset sebagai $Pbest$.
4. Identifikasi partikel dalam lingkungan dengan hasil terbaik sejauh ini.
5. *Update velocity* dan posisi untuk masing-masing partikel.
6. Kembali ke langkah 2 sampai kriteria terpenuhi, biasanya berhenti pada nilai *fitness* yang cukup baik atau sampai pada jumlah maksimum iterasi.

Setiap partikel mempertahankan posisinya, terdiri *fitness* yang telah dievaluasi, dan kecepatannya. Selain itu, setiap partikel mengingat nilai *fitness* terbaik yang telah dicapai sejauh ini selama pengoperasian algoritma, disebut sebagai *fitness* partikel terbaik, dan kandidat solusi yang dicapai oleh *fitness* ini, disebut sebagai yang posisi terbaik partikel ($Pbest$). Akhirnya, algoritma PSO mempertahankan nilai *fitness* terbaik dicapai antara semua partikel dalam *swarm*, yang disebut *fitness* global terbaik, dan kandidat solusi yang dicapai *fitness* ini, disebut posisi terbaik global ($Gbest$).

Algoritma ini terdiri dari tiga langkah, yang diulang sampai kondisi berhenti (Particle Swarm Optimization : A Tutorial, 2009):

1. Mengevaluasi *fitness* dari setiap partikel
2. *Update fitness* partikel dan global terbaik dan posisi
3. *Update* kecepatan dan posisi setiap partikel

Perubahan kecepatan (*velocity*) pada algoritma ini direpresentasikan dengan Persamaan 2-1 (Fukuyama, 2007):

$$v_i^{k+1} = \omega v_i^k + c_1 rand_1 \times Pbest_i - s_i^k + c_2 rand_2 \times Gbest - s_i^k \quad (2.1)$$

dimana :

- v_{ik} : kecepatan agen I pada iterasi k
- ω : fungsi pemberat (inertia)
- c_i : factor pemberat
- $rand$: nilai acak antara 0 dan 1
- s_{ik} : posisi terakhir agen I pada iterasi k
- p_{besti} : best dari agen i
- g_{best} : nilai p_{best} terbaik dari kawanan

Bobot inersia (ω), diperkenalkan oleh Shi dan Eberhart, yang digunakan untuk menyeimbangkan eksplorasi global dan eksploitasi lokal. Bobot inersia *max* memfasilitasi pencarian global, sementara bobot inersia *min* memfasilitasi pencarian lokal. Untuk mengurangi bobot selama iterasi memungkinkan algoritma untuk mengeksplorasi beberapa daerah spesifik (Chen, 2011)

Pada setiap iterasi, nilai fungsi inersia di-update melalui Persamaan 2.2:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \times iter \quad (2.2)$$

Dimana :

ω_{max} : Nilai pemberat (inertia) awal

ω_{min} : nilai pemberat (inertia) akhir

$iter_{max}$: jumlah iterasi maksimum

$iter$: jumlah iterasi terakhir

Seiring dengan berubahnya kecepatan, maka terjadi perubahan pula pada posisi agen di setiap iterasi yang dapat di hitung dengan Persamaan 2.3 : (Fukuyama, 2007):

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (2.3)$$

dimana :

s_i^{k+1} : posisi agen terakhir

s_i^k : posisi agen sebelumnya

v_i^{k+1} : kecepatan agen terkini

Langkah-langkah pada algoritma *Particle swarm optimization* dapat dirangkum dalam pseudocode berikut:

Standard algoritma <i>Particle swarm optimization</i>	
1	Mulai
2	for setiap partikel i
3	inisialisasi posisi dan kecepatan
4	end for
5	while (bukan jumlah iterasi maksimum)
6	for tiap partikel i
7	tentukan <i>fitness</i> value ω_i
8	if ω_i lebih baik dari pbest terkini
9	then pbest <i>fitness</i> = ω_i , pbest = s_i^k
10	if ω_i lebih baik dari gbest terkini
11	then gbest <i>fitness</i> = ω_i , gbest = s_i^k
12	end for
13	for tiap partikel i
14	hitung v_i^k
15	Update v_i^k

16	end for
17	update ω
18	end while
19	End

Gambar 2.1 Pseudo Code Algoritma *Particle swarm optimization*

[Sumber: Nouaouuria, 2005]

2.10 Nilai *Fitness*

Proses perhitungan nilai *fitness* adalah proses evaluasi. Nilai *fitness* untuk digunakan menyatakan nilai dan juga tujuan dan fungsi dari algoritma *particle swarm optimization* adalah untuk memaksimalkan nilai *fitness* tersebut. Nilai *fitness* digunakan untuk menentukan kualitas suatu partikel pada algoritma genetika. Fungsi *fitness* dapat dilakukan dengan Persamaan 2.5.

$$Fitness = C - f(x) \quad (2.4)$$

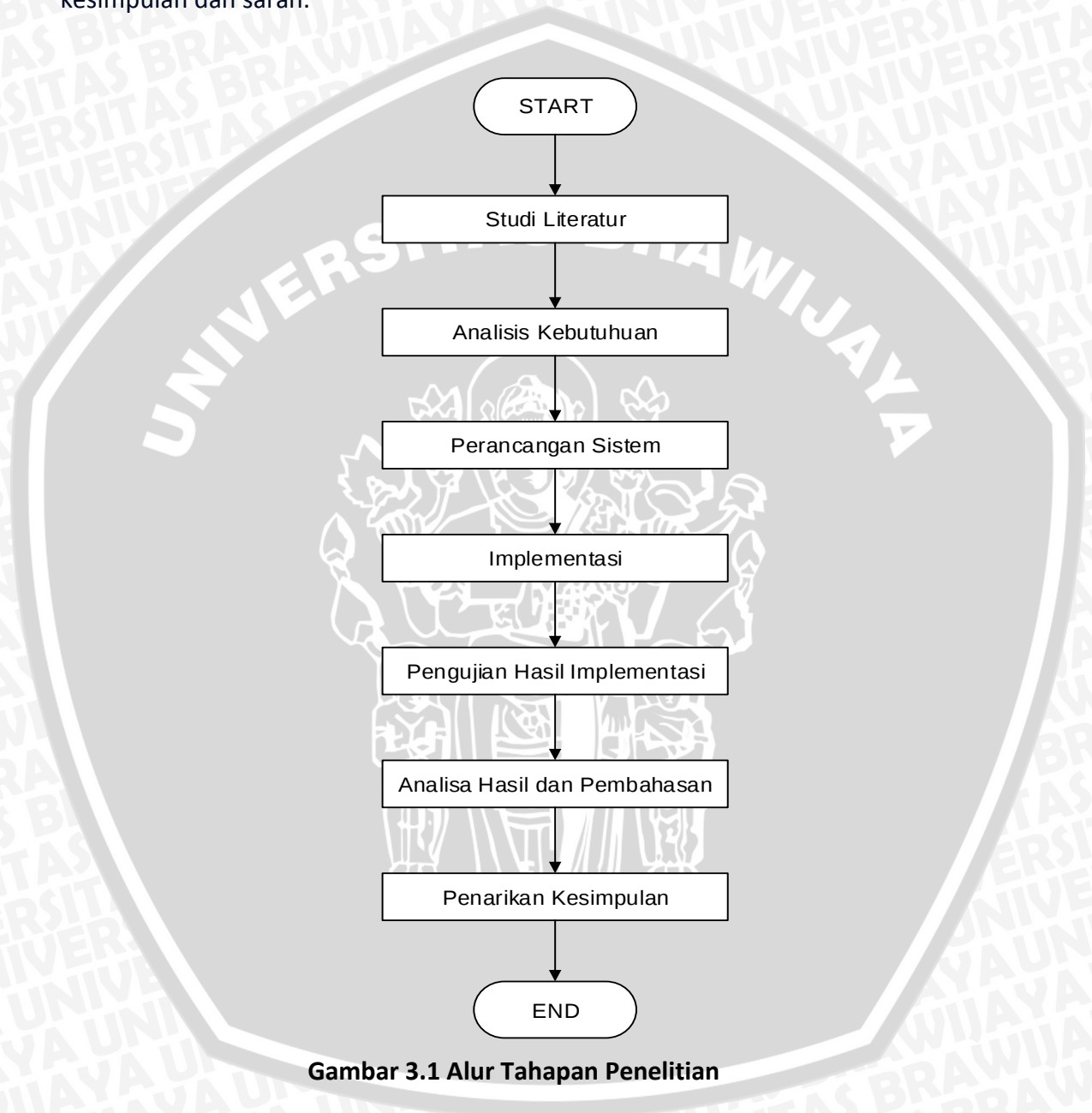
$$Fitness = \frac{c}{f(x) + \varepsilon} \quad (2.5)$$

Keterangan : C = bilangan konstanta
 ε = bilangan kecil untuk menghindari pembagian oleh nol

Pada Persamaan 2.5 ditunjukkan persamaaan nilai *fitness* yang digunakan untuk menyelesaikan sebuah permasalahan yang solusinya adalah memaksimalkan fungsi f (masalah maksimasi), sedangkan Persamaan 2.4 adalah fungsi *fitness* yang digunakan untuk mencari solusi yang meminimalkan fungsi f (Rismayanti, et al., 2015)

BAB 3 METODOLOGI

Bab ini akan membahas langkah-langkah yang akan ditempuh dalam penyusunan skripsi, meliputi studi dan pengkajian literatur, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian, analisis hasil lalu kesimpulan dan saran.



Gambar 3.1 Alur Tahapan Penelitian

3.1 Studi Literatur

Dalam perancangan dan implementasi penelitian ini, perlu dilakukan studi literatur. Literatur berguna sebagai teori pendukung dan landasan dasar dalam penelitian. Teori pendukung tersebut didapat dari buku, jurnal, *paper* dan internet.

Literatur yang digunakan meliputi beberapa objek berikut:

1. Makanan Sehat dan Gizi
2. Angka Kecukupan Gizi
3. Zat Gizi
4. Angka Metabolisme Basal
5. Indeks Massa Tubuh
6. Optimasi
7. Algoritma *Particle swarm optimization*
8. Akurasi Sistem

3.2 Analisis Kebutuhan

3.2.1 Deskripsi Umum Sistem

Secara umum, sistem yang akan dibangun adalah sistem optimasi pemenuhan kebutuhan gizi keluarga dengan menggunakan algoritma *Particle swarm optimization*. Secara garis besar proses berjalannya sistem ini kurang lebih sama seperti yang dijelaskan pada bab sebelumnya. Proses pertama dalam sistem ini adalah input data keluarga berupa nama, jenis kelamin, usia, berat badan, tinggi badan, dan aktifitas. Lalu menginputkan parameter-parameter algoritma *Particle swarm optimization* yaitu ω_{min} , ω_{max} , C_1 , C_2 , iterasi maksimum dan jumlah partikel. Lalu akan dilakukan proses inisialisasi partikel. Setelah itu akan dihitung *fitness* masing-masing partikel dan dilanjutkan dengan mencari *Pbest* dan *Gbest*. Setelah mencapai iterasi maksimum akan dilihat partikel yang memiliki *fitness* terbaik yang menjadi hasil akhir dari proses optimasi ini.

3.2.2 Spesifikasi Kebutuhan Sistem

Analisis kebutuhan dari pengguna dapat dikelompokkan menjadi kebutuhan fungsional dan non fungsional yang akan digunakan dalam perancangan sistem. Analisis kebutuhan sistem berupa:

1. Sistem dapat menerima masukkan data parameter diantaranya berat badan, tinggi badan, umur, biaya dan jumlah anggota keluarga
2. Sistem mampu menampilkan input data parameter dari pengguna.
3. Sistem mampu menampilkan perhitungan hasil Implementasi Algoritma *Particle swarm optimization* untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga
4. Sistem mampu menampilkan hasil rekomendasi optimal.

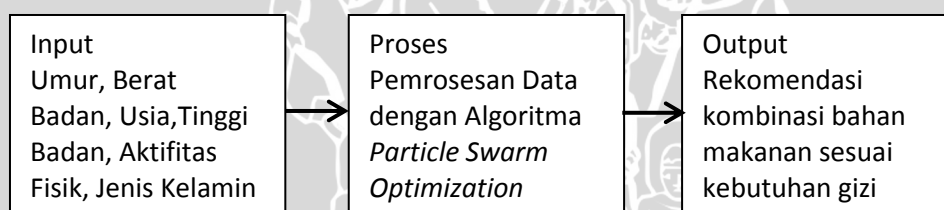
Sedangkan untuk perangkat keras yang dibutuhkan untuk Implementasi Algoritma *Particle swarm optimization* untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga antara lain, *Processor* Intel Core i5-3210M, *Memori* 2 GB 1333 MHz DDR3 2.5GHz, *Hard Drive* (Penyimpanan) 640GB, *Memory* (RAM) 6GB, *Monitor* 15.5 inci dan *Keyboard*. Untuk perangkat lunak yang dibutuhkan antara lain, Sistem Operasi Windows 8, Bahasa pemrograman yang digunakan adalah Java dengan *Tools* pemrograman Netbeans JDK 7.3, DBMS MySQL, dan XAMPP Server 1.8.3-3.

3.2.3 Analisis Kebutuhan Data

Data yang dibutuhkan untuk penelitian ini antara lain, data bahan makanan yang mencakup bahan-bahan makanan itu sendiri, harga bahan makanan per kilogram, ukuran rumah tangga, kalori per bahan makanan, dan kandungan zat-zat gizi pada masing-masing bahan makanan. Data anjuran porsi sehari untuk proses perhitungan takaran setiap bahan makanan untuk masing-masing individu. Data harga makanan yang digunakan untuk perhitungan harga total dari hasil rekomendasi bahan makanan. Setelah itu ada data konsumsi keluarga dalam jangka waktu satu hari untuk dijadikan perbandingan dengan hasil rekomendasi dari sistem.

3.3 Perancangan Sistem

Perancangan sistem membahas tahap-tahap yang akan dilakukan oleh sistem untuk mengimplementasikan metode yang dipakai dan juga perhitungan manual atau manualisasi dari metode tersebut. Perangkat lunak memiliki bagian sistem yang memproses data input pengguna untuk menghasilkan *output* berupa rekomendasi bahan makanan optimal yang memenuhi kebutuhan gizi keluarga. Perancangan sistem dapat dilakukan setelah proses pengumpulan data selesai dilakukan agar sistem dapat menerapkan Implementasi Algoritma *Particle swarm optimization* untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga sehingga dari perancangan tersebut dapat diimplementasikan pada tahap selanjutnya.



Gambar 3.2 Gambaran Umum Perancangan Sistem

3.4 Implementasi

Implementasi adalah tahap realisasi atau menerapkan rancangan yang telah dibuat di tahap perancangan di atas yaitu menerapkan Implementasi Algoritma *Particle swarm optimization* untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga. Implementasi akan dilakukan dengan menggunakan bahasa pemrograman Java dan *database* MySQL.

3.5 Pengujian Hasil Implementasi

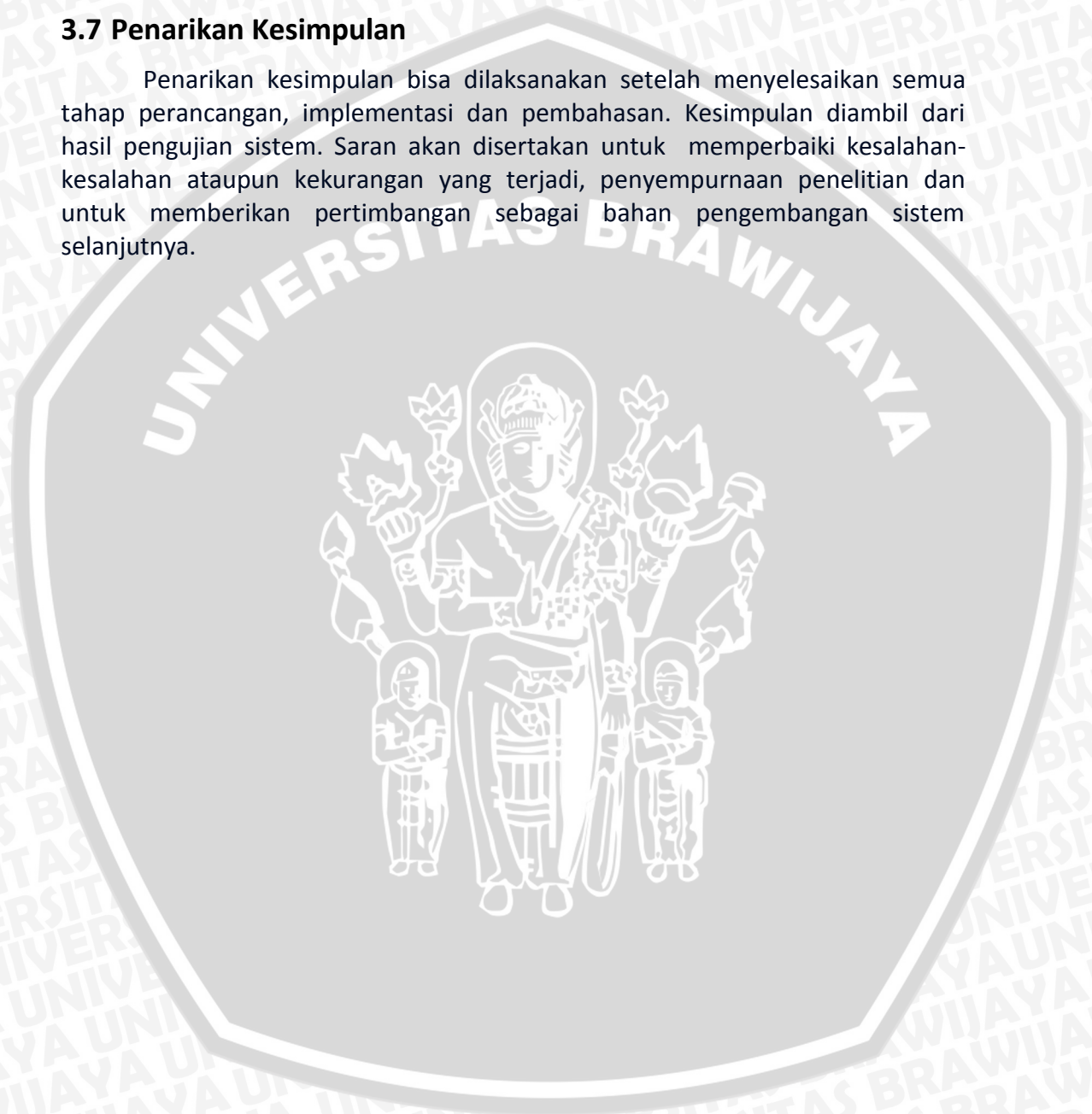
Pengujian hasil implementasi adalah tahap dimana hasil dari implementasi dibandingkan dengan data hasil manualisasi yang telah dibuat dalam tahap perancangan, pada tahap ini dilihat perbandingan presentasi akurasi antara sistem yang telah dibuat dan hasil manualisasi dan juga pengujian-pengujian parameter yang berpengaruh terhadap hasil akhir dari sistem.

3.6 Analisis Hasil dan Pembahasan

Hasil dan pembahasan akan menjelaskan hasil dari tahap implementasi di atas yaitu Implementasi Algoritma *Particle swarm optimization* untuk Optimasi Pemenuhan Kebutuhan Gizi Keluarga dan juga menguji metode yang telah diimplementasikan.

3.7 Penarikan Kesimpulan

Penarikan kesimpulan bisa dilaksanakan setelah menyelesaikan semua tahap perancangan, implementasi dan pembahasan. Kesimpulan diambil dari hasil pengujian sistem. Saran akan disertakan untuk memperbaiki kesalahan-kesalahan ataupun kekurangan yang terjadi, penyempurnaan penelitian dan untuk memberikan pertimbangan sebagai bahan pengembangan sistem selanjutnya.



BAB 4 PERANCANGAN

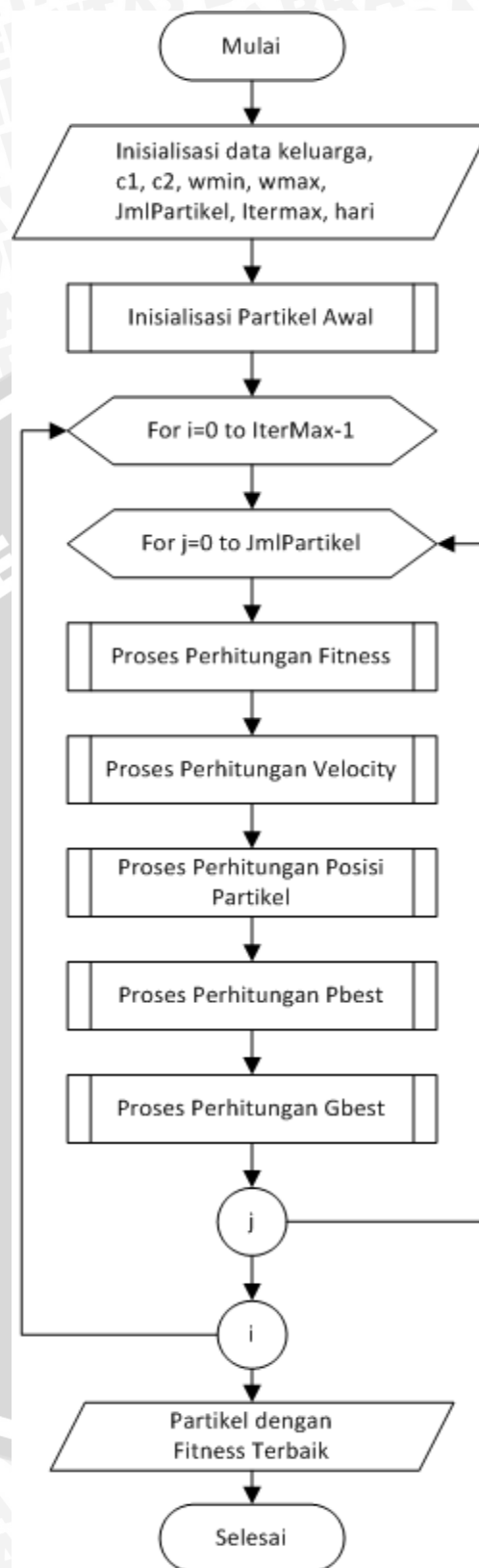
Pada bab ini akan dijabarkan tahapan-tahapan perancangan dari sistem ini. Mulai dari simulasi penyelesaian masalah secara sederhana, perancangan antarmuka untuk pengguna dan perancangan pengujian yang akan dilakukan terhadap sistem. Dijelaskan pada Gambar 4.1 yang merupakan diagram alir proses optimasi menggunakan algoritma *particle swarm optimization* secara umum.

4.1 Penyelesaian Masalah Menggunakan Algoritma *Particle swarm optimization*

Pada sub bab ini akan dibahas mengenai langkah-langkah penyelesaian masalah optimasi pemenuhan gizi keluarga dengan menggunakan algoritma *Particle swarm optimization*. Gambar 4.1 dibawah menunjukkan proses kerja secara umum Algoritma *Particle swarm optimization* dalam permasalahan optimasi pemenuhan gizi keluarga.

Berikut adalah langkah-langkah kerja algoritma yang akan dijelaskan lebih lanjut.

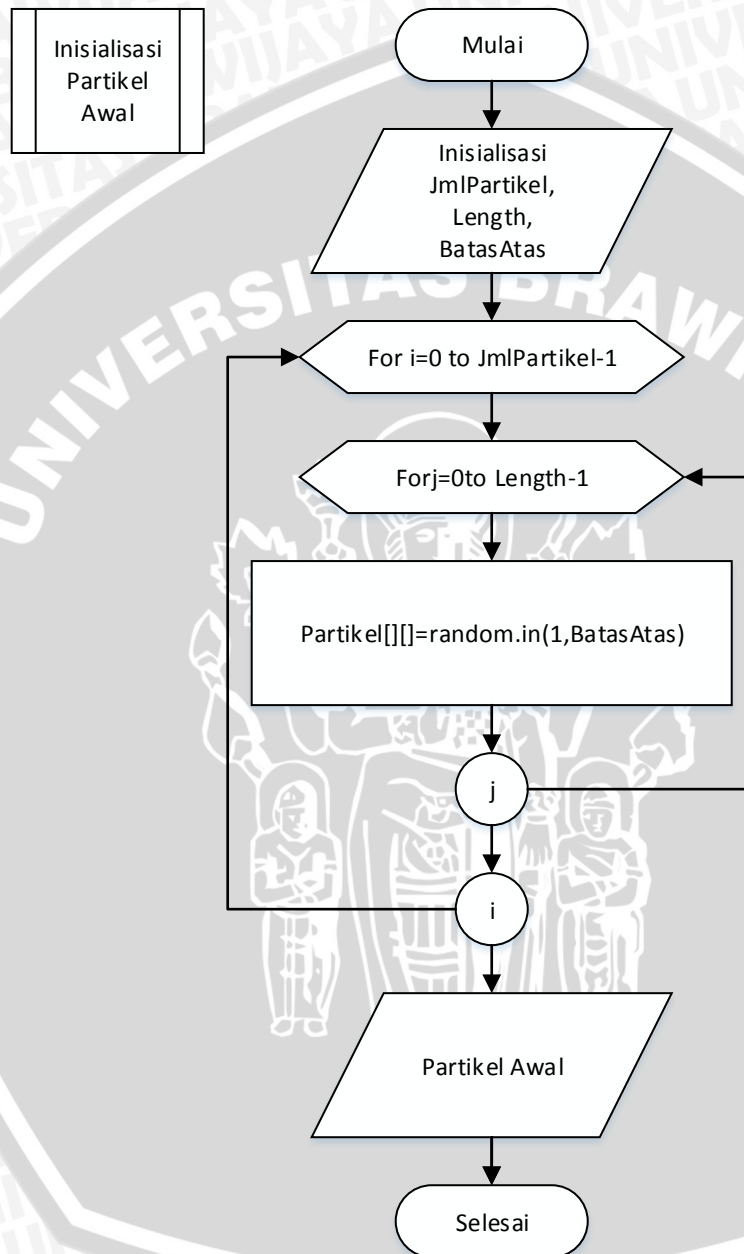
1. Menerima inputan-inputan parameter diantaranya parameter untuk menentukan kebutuhan gizi masing-masing anggota keluarga yaitu parameter jumlah anggota keluarga, berat badan, tinggi badan, usia, aktivitas dan jumlah hari. Untuk parameter algoritma *Particle swarm optimization* yang diinputkan adalah ω_{min} , ω_{max} , C_1 , C_2 , iterasi maksimal yang disimpan dalam variabel IterMax, dan jumlah partikel yang disimpan dalam variabel JmlPartikel .
2. Proses memulai inialisasi *Swarm* atau populasi partikel awal.
3. Proses menghitung *fitness* masing-masing partikel
4. Proses penentuan *Pbest* dan *Gbest* dari setiap iterasi
5. Proses menghitung kecepatan setiap partikel untuk iterasi selanjutnya
6. Proses menghitung posisi partikel untuk iterasi selanjutnya
7. Lalu kembali ke penghitungan *fitness* dan proses akan berlanjut hingga sampai pada jumlah iterasi maksimum dan jumlah partikel.



Gambar 4.1 Proses Algoritma *Particle Swarm Optimization* dalam Optimalisasi Pemenuhan Gizi Secara Umum

4.1.1 Proses inialisasi awal

Untuk inialisasi partikel yang digunakan adalah angka permutasi dengan range yang akan ditentukan pada proses yang lalu akan dikonversi menjadi *indeks* bahan makanan pada proses perhitungan *fitness*. Langkah-langkah yang dilakukan dalam proses inialisasi awal sesuai dengan Gambar 4.2.



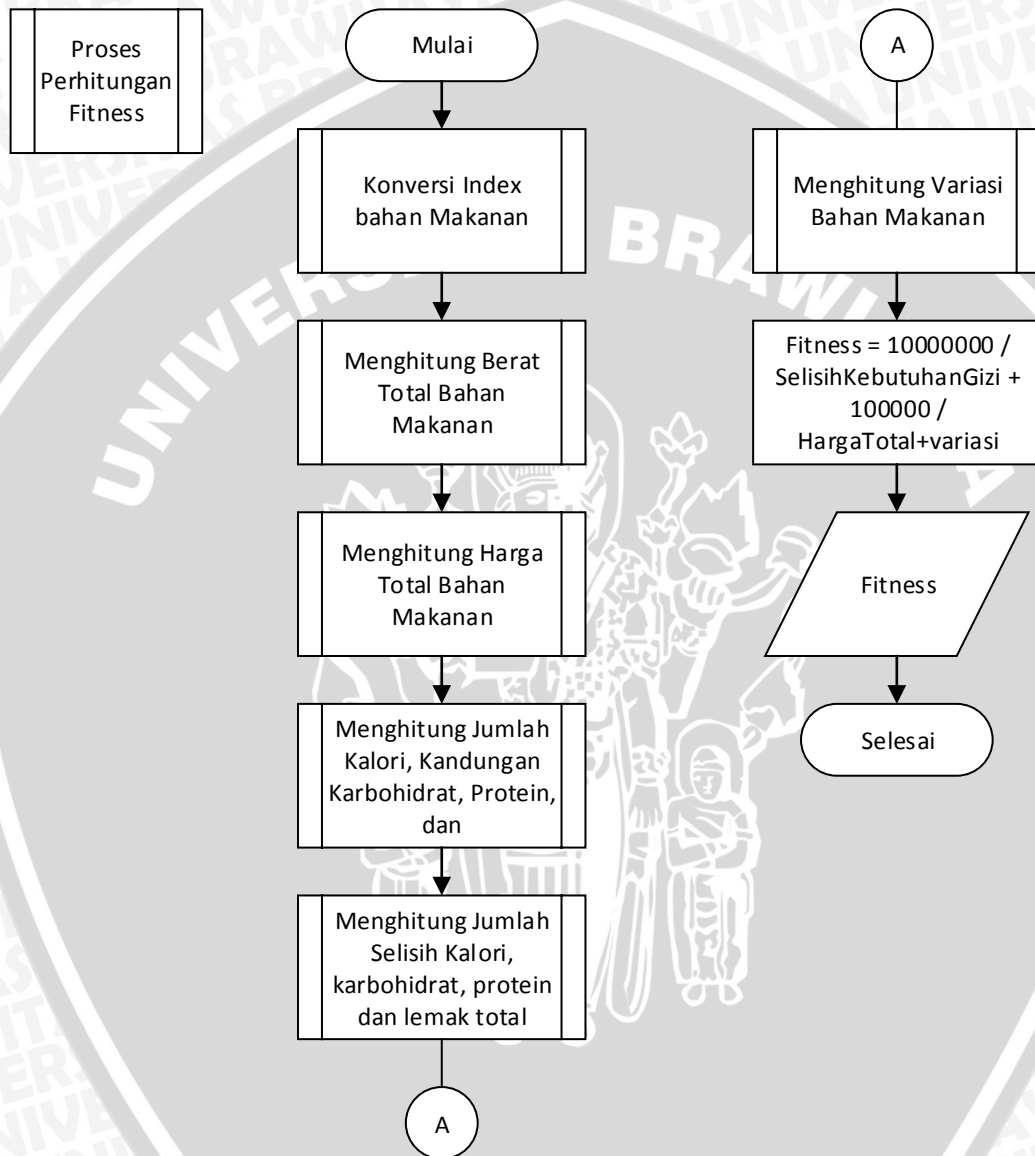
Gambar 4.2 Proses Inisialisasi Partikel Awal

Langkah-langkah yang terjadi dalam proses inialisasi partikel awal sesuai dengan Gambar 4.2 adalah sebagai berikut :

1. Memasukkan nilai JmlPartikel dan Length.
2. Memulai perulangan sebanyak Length.

3. Membangkitkan bilangan acak sepanjang Length. Proses diulang sejumlah JmlPartikel.
4. Mengembalikan *array* berisi partikel awal.
5. Selesai.

4.1.2 Proses Perhitungan *Fitness*



Gambar 4.3 Proses Penghitungan *Fitness*

Langkah-langkah yang dilakukan dalam proses perhitungan nilai *fitness* sesuai *flowchart* pada Gambar 4.3 adalah sebagai berikut:

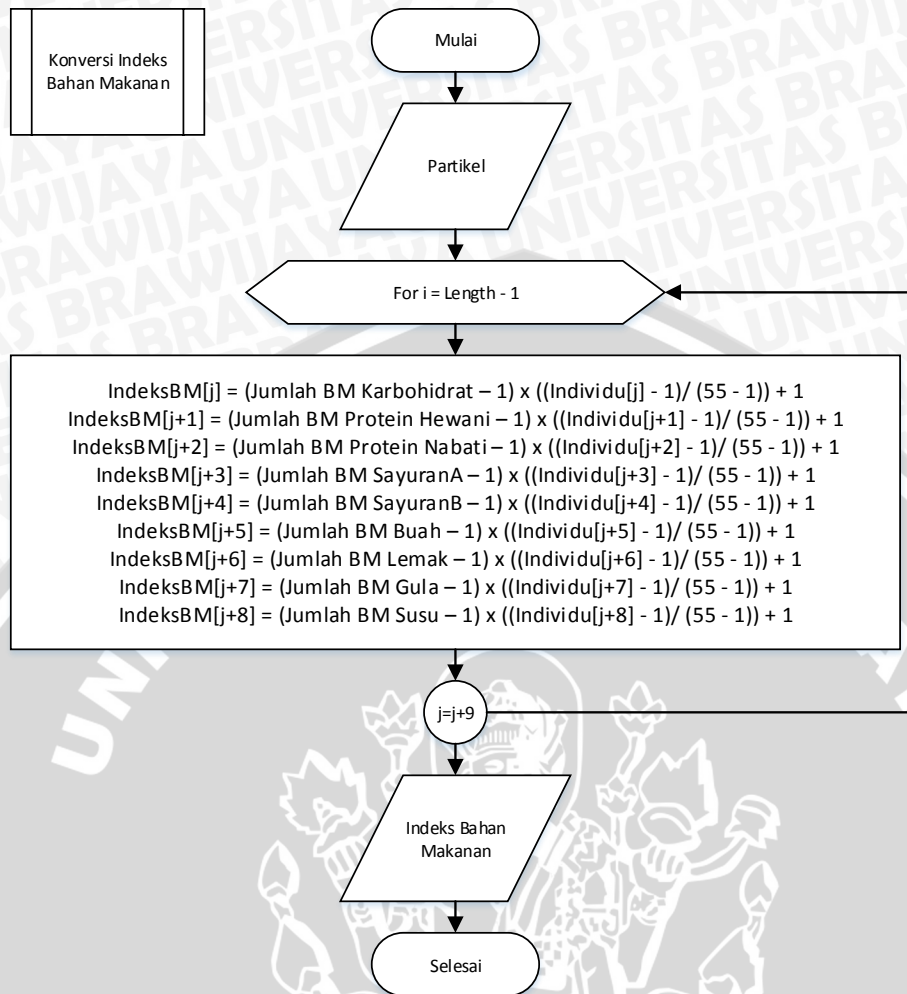
1. Mengisi *array* partikel.
2. Mengkonversi angka permutasi dalam partikel menjadi indeks bahan makanan yang sesuai dalam tabel .

3. Mengkonversi indeks bahan makanan menjadi nama bahan makanan yang sesuai dalam tabel.
4. Menghitung takaran masing-masing bahan makanan untuk setiap individu anggota keluarga.
5. Menghitung berat total setiap bahan makanan untuk seluruh keluarga.
6. Menghitung harga total semua bahan makanan .
7. Menghitung kalori masing-masing bahan makanan jatah setiap anggota keluarga
8. Menghitung selisih antara kalori yang dikonsumsi dengan jumlah energi yang dibutuhkan.
9. Menghitung jumlah karbohidrat yang terkandung dalam masing-masing bahan makanan yang dikonsumsi setiap individu anggota keluarga
10. Menghitung selisih antara karbohidrat yang dikonsumsi dengan jumlah karbohidrat yang dibutuhkan.
11. Menghitung jumlah protein yang terkandung dalam masing-masing bahan makanan yang dikonsumsi setiap individu anggota keluarga
12. Menghitung selisih antara protein yang dikonsumsi dengan jumlah protein yang dibutuhkan.
13. Menghitung jumlah lemak yang terkandung dalam masing-masing bahan makanan yang dikonsumsi setiap individu anggota keluarga
14. Menghitung selisih antara lemak yang dikonsumsi dengan jumlah kebutuhan lemak.
15. Menghitung variasi bahan makanan dalam satu partikel.
16. Menghitung nilai *fitness*.
17. Mengembalikan nilai *fitness*.
18. Selesai.

4.1.2.1 Proses Konversi Indeks Bahan Makanan

Langkah-langkah yang dilakukan dalam proses konversi angka permutasi menjadi indeks bahan makanan sesuai *flowchart* pada Gambar 4.4 adalah sebagai berikut :

1. Memasukkan partikel yang akan diproses.
2. Melakukan konversi dengan perhitungan sesuai Persamaan 4-1.
3. Mengembalikan *array IndeksBM*.

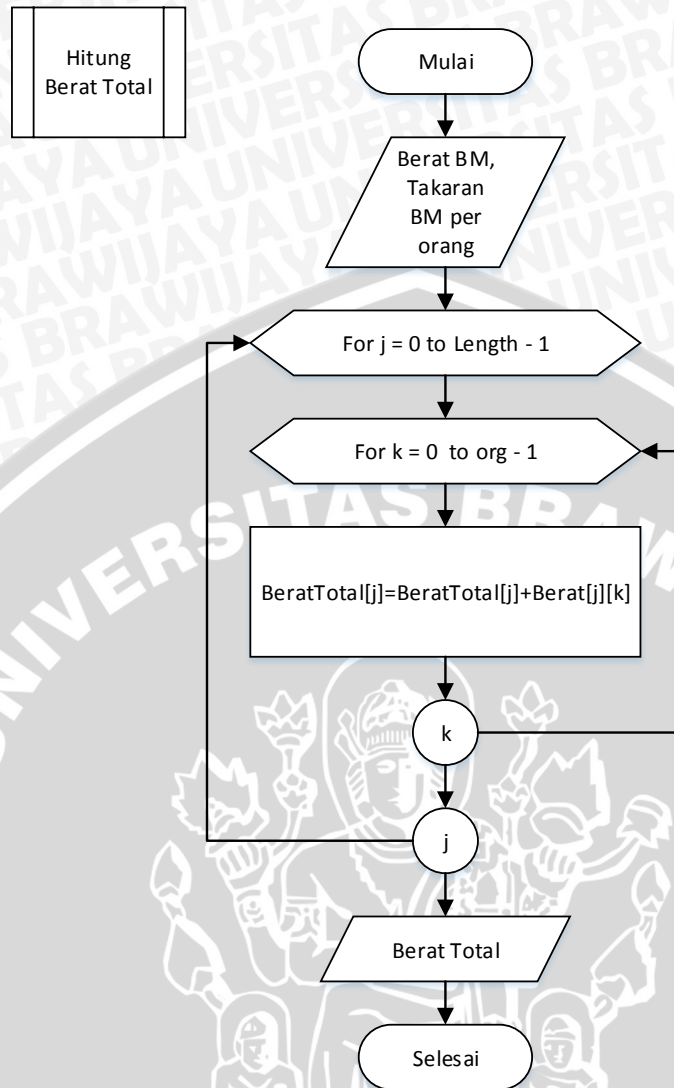


Gambar 4.4 Proses Konversi Indeks Bahan Makanan

4.1.2.2 Proses Hitung Berat Total

Langkah-langkah yang dilakukan dalam proses perhitungan berat bahan makanan sesuai *flowchart* pada Gambar 4.5 adalah sebagai berikut:

1. Inisialisasi berat masing-masing bahan makanan dan juga takaran bahan makanan per orang.
2. Memulai perulangan sebanyak panjang partikel.
3. Menghitung berat total dengan menjumlah berat bahan makanan untuk setiap orang dalam satu keluarga.
4. Mengakhiri perulangan.
5. Mengembalikan variabel BeratTotal.

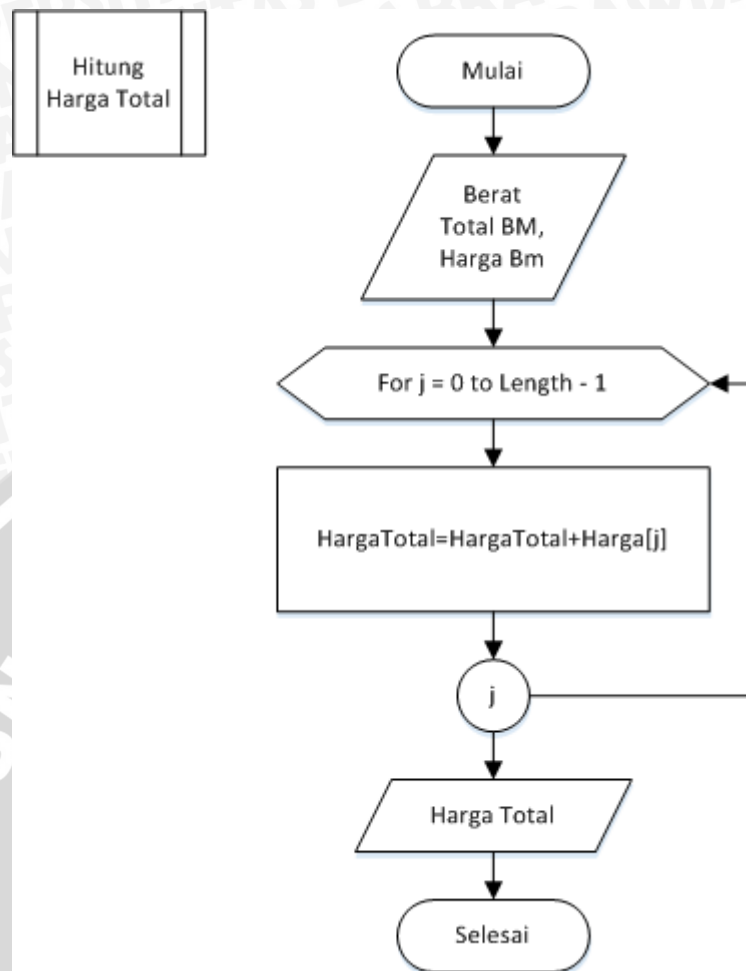


Gambar 4.5 Proses Hitung Berat Total

4.1.2.3 Proses Hitung Harga Total

Langkah-langkah yang dilakukan dalam proses perhitungan harga total sesuai *flowchart* pada Gambar 4.6 adalah sebagai berikut:

1. Inisialisasi panjang gen dan variabel penyimpan hasil proses ini (HargaTotal).
2. Memulai perulangan sebanyak jumlah gen.
3. Menjumlahkan harga setiap bahan makanan dalam satu individu.
4. Mengakhiri perulangan.
5. Mengembalikan variabel HargaTotal

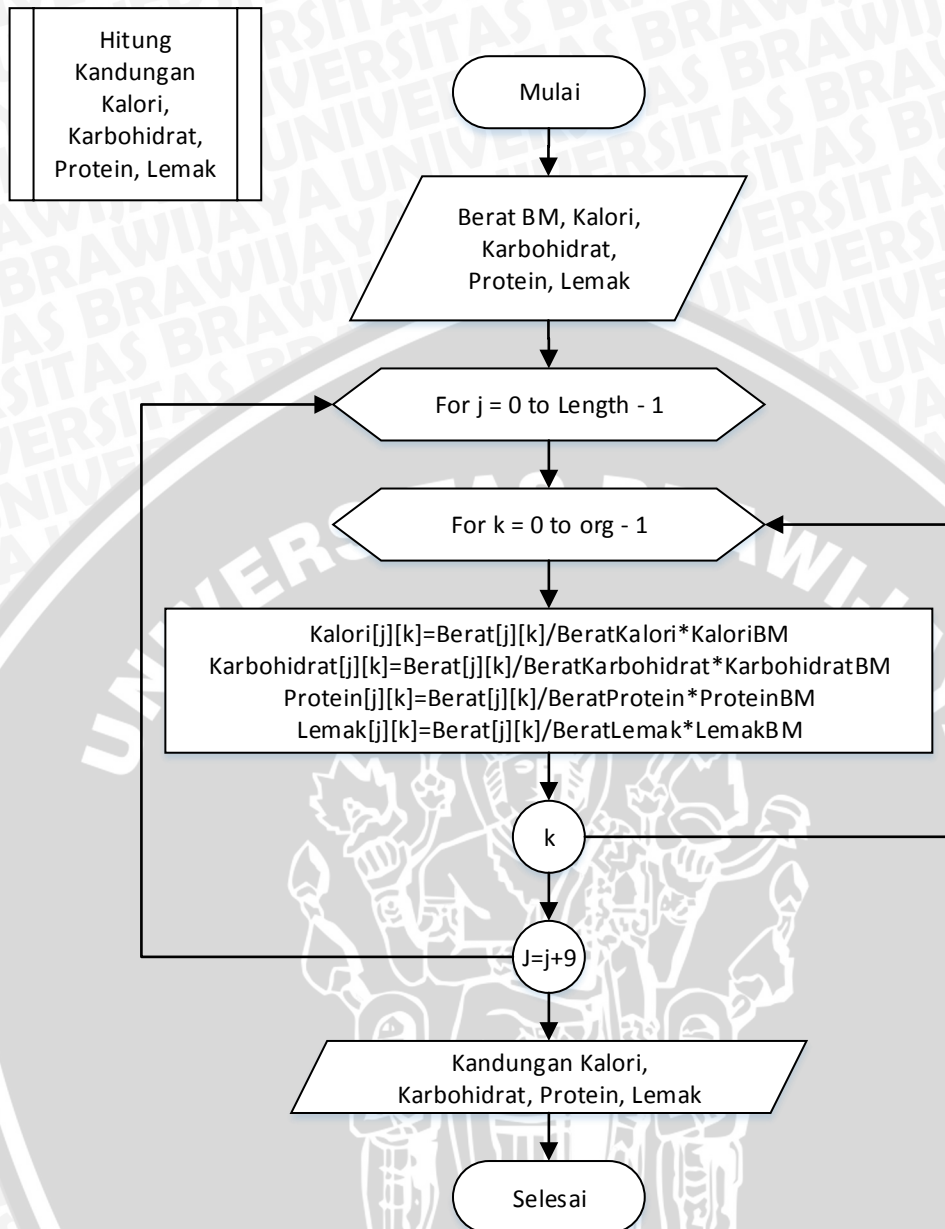


Gambar 4.6 Proses Hitung Harga Total

4.1.2.4 Proses Perhitungan Kandungan Gizi

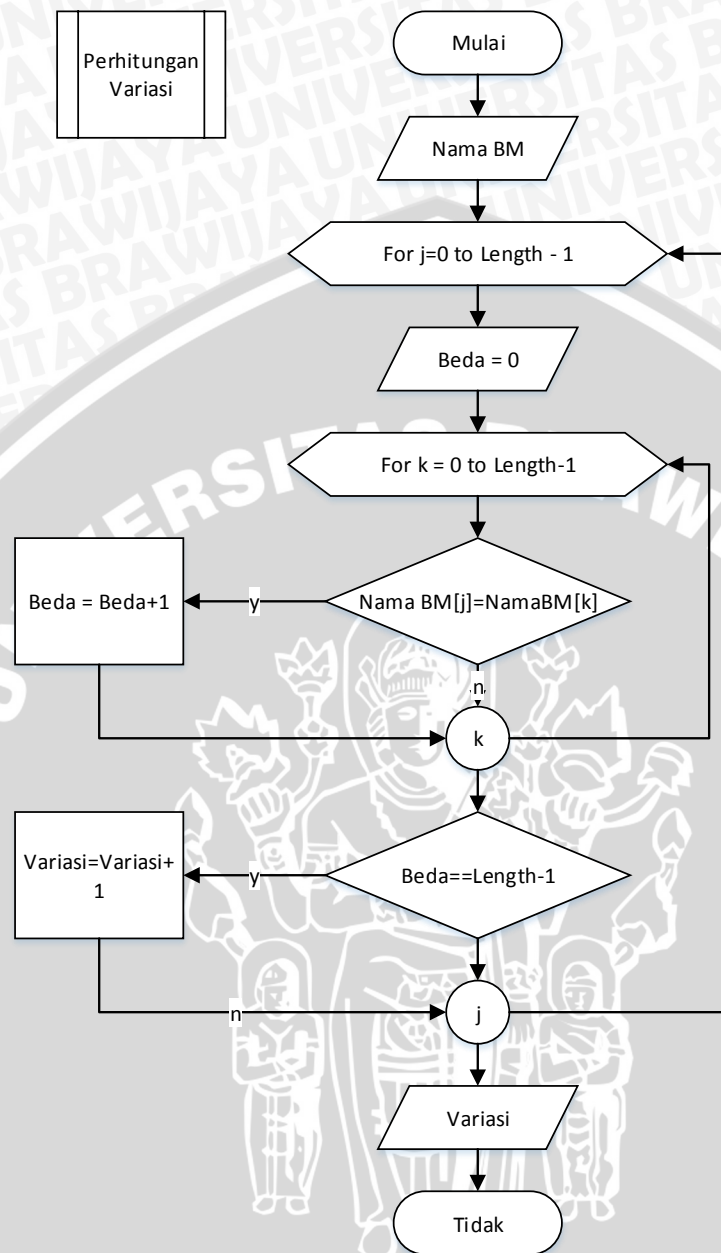
Langkah-langkah yang dilakukan dalam proses perhitungan kandungan kalori sesuai *flowchart* pada Gambar 4.7 adalah sebagai berikut:

1. Inisialisasi berat bahan makanan, jumlah kalori, kandungan karbohidrat, protein, dan lemak.
2. Memulai perulangan sebanyak panjang partikel
3. Memulai perulangan sebanyak jumlah orang.
4. Menghitung kandungan kalori, karbohidrat, protein, dan lemak sesuai berat bahan makanan untuk masing-masing anggota keluarga.
5. Mengakhiri semua perulangan.
6. Mengembalikan variabel kalori, karbohidrat, protein, dan lemak.



Gambar 4.7 Proses Perhitungan Kandungan Gizi

4.1.2.5 Proses Perhitungan Variasi



Gambar 4.8 Proses Perhitungan Variasi

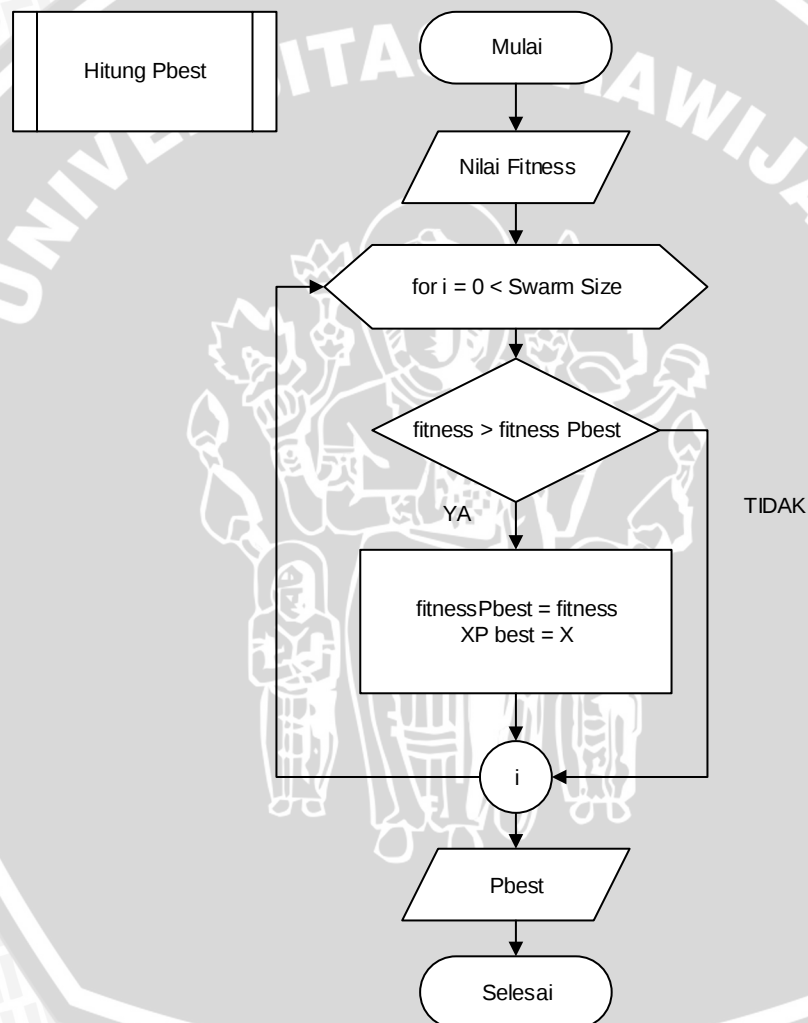
Langkah-langkah yang dilakukan dalam proses perhitungan variasi sesuai *flowchart* pada Gambar 4.8 adalah sebagai berikut:

1. Inisialisasi nama bahan makanan.
2. Memulai perulangan sebanyak panjang partikel Length.
3. Memberi nilai variabel beda dengan nilai 0.
4. Memulai perulangan sebanyak panjang partikel Length.
5. Membandingkan nama bahan makanan satu sama lain dalam satu partikel. Jika sama maka nilai variabel beda bertambah 1.

6. Mengulang langkah ke-5 hingga indeks Length terakhir.
7. Jika nilai variabel beda sama dengan jumlah length dikurangi 1, maka nilai variabel variasi bertambah 1.

4.1.3 Proses Perhitungan Pbest dan Gbest

Pada setiap iterasi, akan selalu dilakukan pembaruan nilai *Pbest*, nilai *Pbest* didapat dengan membandingkan nilai *fitness* setiap partikel pada setiap iterasi, pada dasarnya *fitness* antara partikel ke-*i* pada iterasi ke-*j* akan dibandingkan dengan *fitness* partikel ke-*i* dengan iterasi ke-*(j-1)*, Partikel yang menghasilkan *fitness* terbaik pada perbandingan tersebut akan dijadikan *Pbest*. Tahapan proses pengecekan dan pembaruan nilai *Pbest* akan dijelaskan pada Gambar 4.9

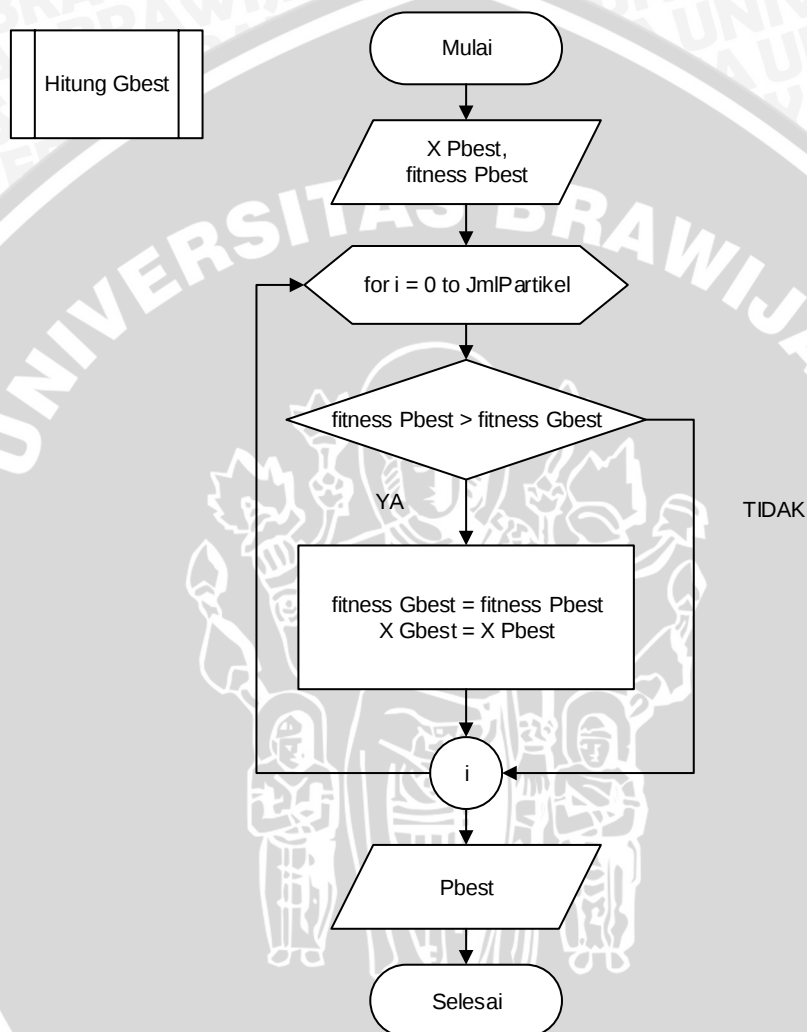


Gambar 4.9 Proses Perhitungan Pbest

1. Nilai *Fitness* yang sudah dihitung dimasukkan dalam proses penghitungan *Pbest*
2. Lakukan pengecekan jika $fitness < fitness_{pbest}$, maka $X_{pbest} = X$ dan $fitness_{pbest} = fitness$. Hal ini menyatakan jika nilai *fitness* yang dihasilkan pada iterasi

tersebut memiliki nilai yang lebih besar daripada *fitness* pbest sebelumnya maka nilai *fitness* terakhir yang dihasilkan adalah nilai *fitness* terbaik yang pernah didapat selama proses optimasi. Hasil dari proses ini adalah nilai *Pbest* dari partikel.

Sedangkan untuk nilai *Gbest* didapatkan dari *fitness* terbaik yang terdapat pada sebuah *swarm* atau populasi. Tahapan proses pencarian nilai *Gbest* dijelaskan pada Gambar 4.10

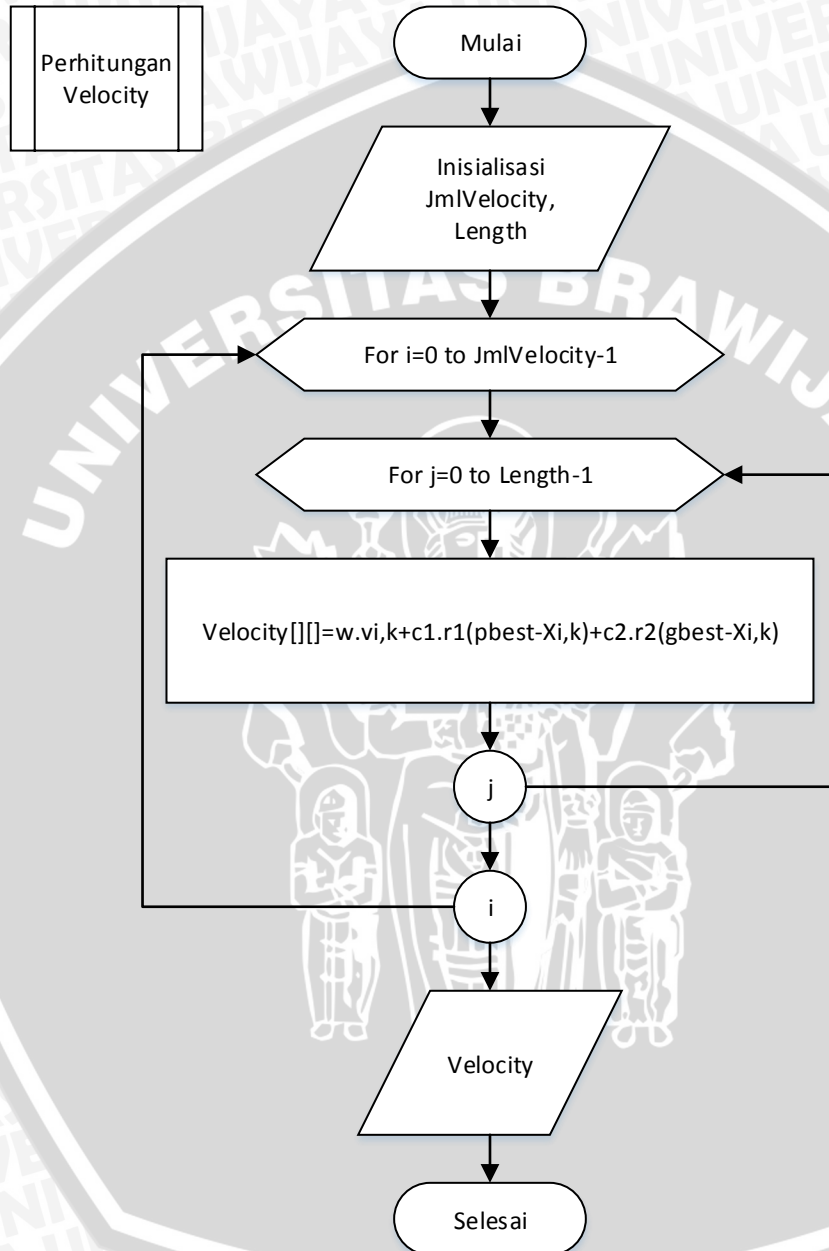


Gambar 4.10 Proses Perhitungan Gbest

Nilai *Xpbest* dan *fitnesspbest* yang merupakan posisi terbaik yang pernah dicapai oleh masing-masing partikel beserta nilainya di-input-kan dalam proses pencarian *Gbest*. Lalu pada iterasi dari nilai *i* =1 hingga *jmlPartikel* dilakukan pengecekan jika *fitnesspbest* < *fitnessgbest*, maka ,nilai *fitnesspbest* adalah nilai *fitnessgbest* terbaru. Nilai *Xpbest* adalah nilai *Xgbest* terbaru. Hasil dari proses ini adalah nilai *gbest*.

4.1.4 Proses Perhitungan Perubahan Kecepatan (*Velocity*)

Perhitungan perubahan kecepatan dilakukan untuk merubah kecepatan masing-masing partikel pada setiap iterasi. Hal menyatakan kemana arah partikel-partikel tersebut akan terbang. Dapat dilihat proses perhitungan perubahan kecepatan pada Gambar 4.11



Gambar 4.11 Proses Perhitungan *Velocity*

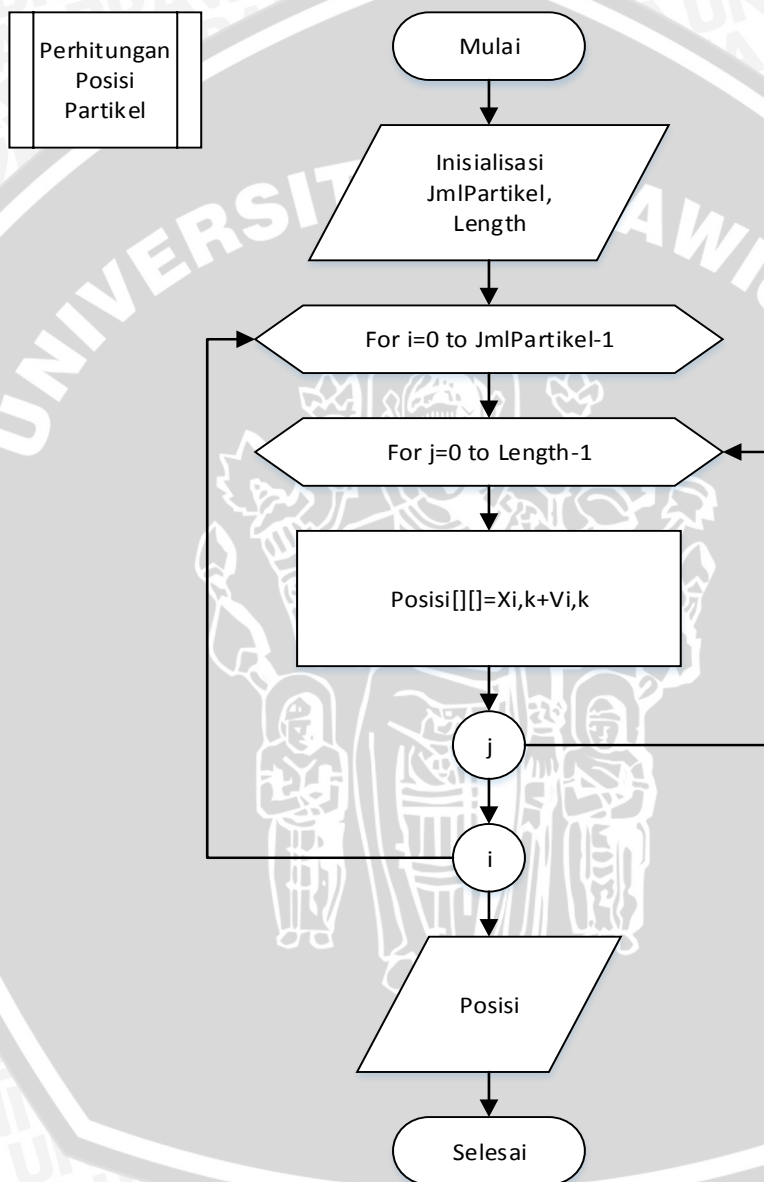
Langkah-langkah yang dilalui dalam proses perhitungan perubahan kecepatan sesuai dengan Gambar 4.11 adalah sebagai berikut:

1. Memasukkan nilai *Jmlvelocity* dan *Length*.
2. Memulai perulangan sebanyak *Length*.

3. Menghitung *velocity*. Proses diulang sejumlah *Jmlvelocity*.
4. Mengembalikan *array* berisi nilai *velocity* partikel.
5. Selesai.

4.1.5 Proses Perhitungan Perubahan Posisi Partikel

Bersamaan dengan terjadinya perubahan kecepatan, maka posisi masing-masing partikel juga tentunya akan turut berubah pada setiap iterasi.



Gambar 4.12 Proses Perhitungan Posisi Partikel

Langkah-langkah yang dilalui dalam proses perhitungan perubahan kecepatan sesuai dengan Gambar 4.12 adalah sebagai berikut:

1. Memasukkan nilai *JmlPartikel* dan *Length*.
2. Memulai perulangan sebanyak *Length*.

3. Menghitung velocity. Proses diulang sejumlah JmlPartikel.
4. Mengembalikan *array* berisi posisi partikel
5. Selesai

4.2 Perhitungan Manual

Misalkan Data Keluarga yang diperoleh adalah:

Tabel 4.1 Tabel Data Anggota Keluarga

Data Anggota Keluarga							
No	Nama	Usia	Jenis Kelamin	Berat Badan	Tinggi Badan	Aktivitas	Kebutuhan Energi (kkal)
1	Ayah	51	L	76	160	Sedang	2745
2	Ibu	49	P	72	155	Ringan	2178
3	Anak 1	21	P	52	158	Sedang	2278

Maka untuk perhitungan kebutuhan energi setiap individu adalah sebagai berikut:

$$\text{Ayah} = 65 + (13,7 * 76) + (5 + 160) - (6,8 * 51)) * 1,76 = 2745 \text{ kkal}$$

$$\text{Ibu} = 665 + (9,6 * 72) + (1,8 + 155) - (4,7 * 49)) * 1,55 = 2178 \text{ kkal}$$

$$\text{Anak 1} = 665 + (9,6 * 52) + (1,8 + 158) - (4,7 * 21)) * 1,70 = 2278 \text{ kkal}$$

4.2.1 Parameter Algoritma *Particle Swarm Optimization*

Parameter yang harus diinisialisasi pada proses optimasi dengan algoritma *particle swarm optimization* adalah ω_{min} , ω_{max} , C_1 , C_2 , dan IterMax atau iterasi maksimal dan JmlPartikel atau jumlah partikel. Pada perhitungan manual kali ini, nilai dari parameter tersebut adalah:

$$\omega_{min} = 0.4$$

$$\omega_{max} = 0.9$$

$$C_1 = 2$$

$$C_2 = 2$$

$$\text{IterMax} = 2$$

$$\text{JmlPartikel} = 2$$

4.2.2 Inisialisasi Partikel

Pada penelitian ini data yang digunakan adalah 178 bahan makanan yang dibagi menjadi 9 jenis yaitu, bahan makanan sumber karbohidrat, protein hewani, nabati, sayuran jenis A, sayuran jenis B, buah, gula, lemak, dan susu. Satu bahan makanan diwakili oleh satu agen dalam partikel. Sedangkan dalam setiap partikel terdiri dari 189 agen yang didapat dari 9 jenis bahan makanan, 7 hari dan 3 kali makan dalam sehari. Partikel-partikel ini nantinya akan menjadi

dasar untuk menentukan kombinasi bahan makanan untuk 7 hari dengan frekuensi makan setiap harinya sebanyak 3 kali.

Untuk perhitungan manual, diambil sampel sepanjang 24 agen. 24 agen tersebut menjadi dasar untuk menentukan kombinasi bahan makanan untuk 2 hari yang terdiri dari 4 jenis bahan makanan yaitu bahan makanan sumber karbohidrat, protein hewani, protein nabati, dan lemak. Bilangan bulat dengan rentang 1 sampai 55 yang diambil acak digunakan untuk masing-masing agen. 1-55 ini adalah batas angka permutasi yang ditentukan agar mempermudah pembentukan partikel dan menjalankan proses algoritma. Jika partikel langsung menggunakan indeks bahan makanan, setiap agen harus diisi dengan *range* yang berbeda sesuai dengan jumlah masing-masing jenis bahan makanan dimana hal ini akan menyulitkan jalannya proses algoritma, karena bisa saja angka yang dihasilkan pada iterasi selanjutnya melebihi ketentuan *range*. Karena diambil secara random, angka 1-55 ini nantinya akan dinormalisasi agar sesuai dengan indeks bahan makanan yang ada. Gambar 4.13 menunjukkan representasi partikel pada penelitian ini.

Partikel ke-	Hari ke-1												Hari ke-2											
	Pagi				Siang				Malam				Pagi				Siang				Malam			
	KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L	KH	PH	PN	L
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	Partikel																							
X1	11	24	19	6	23	31	50	7	42	39	26	39	46	3	34	20	24	26	45	35	28	23	39	25
X2	29	40	21	9	22	55	31	8	3	39	23	54	52	16	4	34	37	46	12	23	35	3	32	12

Gambar 4.13 Inisialisasi Partikel Awal

Keterangan:

KH : Karbohidrat

PH : Protein hewani

PN : Protein nabati

L : Lemak

Bilangan bulat 1-55 tadi nantinya akan di normalisasi dengan persamaan dibawah untuk penentuan indeks bahan makanan sesuai jenis bahan makanan.

$$Indeks\ gen\ ke - i = x - \frac{y-1}{z-1} + 1 \quad (4-1)$$

Keterangan :

x : Jumlah anggota jenis bahan makanan ke- i

y : Nilai gen ke- j

z : Batas atas angka permutasi

Sebagai contoh, perhitungan untuk agen indeks ke 15 partikel X1 dengan nilai 34.

$$X_{1,15} = 14 - \frac{34 - 1}{55 - 1} + 1 = 8$$

Dari hasil contoh perhitungan ditunjukkan bahwa bahan makanan yang dihasilkan adalah sumber protein nabati. Ditunjukkan pada Tabel 4.2 data bahan makanan sumber protein nabati. Pada tabel dapat dilihat indeks 8 adalah kembang tahu.

Tabel 4. 2 Tabel Hasil Konversi Indeks Bahan Makanan

Kode	Bahan Makanan	Harga per kg	URT	Berat (g)	Kalori
1	Kacang Hijau	19000	2 sdm	20	75
2	Kacang Kedele	10000	2 1/2 sdm	25	75
3	Kacang Merah	24000	2 sdm	20	75
4	Kacang Mente	145000	2 1/2 sdm	15	75
5	Kacang Tanah	22000	2 sdm	15	75
6	Kacang Tanah Kupas	22000	2 sdm	15	75
7	Kacang Tolo	15000	2 sdm	20	75
8	Kembang Tahu	165000	1 lembar	20	75
9	Oncom	35000	2 ptg kecil	40	75
10	Pete Segar	10000	1/2 gls	55	75
11	Selai Kacang Tanah	128000	1 sdm	15	75
12	Tahu	10000	1 bj besar	110	75
13	Tempe	10000	2 ptg sdg	50	75

Tabel 4.3 menunjukkan hasil konversi indeks bahan makanan seperti contoh perhitungan di atas menjadi nama bahan makanan.

Tabel 4. 3 Tabel Hasil Konversi Nama Bahan Makanan

Hari ke-	Waktu Makan	Jenis Bahan Makanan	Gen	Indeks Bahan Makanan	Nama Makanan
1	Pagi	KH	11	4	Kentang
		PH	24	18	Putih telur ayam
		PN	19	5	Kacang Tanah
		L	6	2	Kacang Almond
	Siang	KH	23	8	Mi Kering
		PH	31	23	Udang segar
		PN	50	12	Tahu
		L	7	2	Kacang Almond
	Malam	KH	42	13	Roti Gandum
		PH	39	28	Hati sapi
		PN	26	7	Kacang Tolo

2	Pagi	L	39	9	Kelapa
		KH	46	14	Singkong
		PH	3	2	Cumi-cumi
		PN	34	8	Kembang Tahu
	Siang	L	20	5	Minyak Kacang Kedele
		KH	24	8	Mi Kering
		PH	26	19	Rebon kering
		PN	45	11	Selai Kacang Tanah
	Malam	L	35	8	Santan
		KH	28	9	Nasi Beras Giling
		PH	23	17	Kerang
		PN	39	9	Oncom
		L	25	6	Minyak Zaitun

4.2.3 Perhitungan Total Harga

Untuk perhitungan harga total langkah pertama adalah menghitung harga setiap bahan makanan sesuai dengan berat keseluruhan. Contoh perhitungan harga pada partikel X1 untuk porsi makan pagi adalah sebagai berikut.

$$\text{Kentang} = \frac{997,5}{1000} \times 43715 = \text{Rp } 11970,00$$

$$\text{Putih Telur} = \frac{146,25}{1000} \times 18000 = \text{Rp } 2632,50$$

$$\text{Kacang tanah} = \frac{33,75}{1000} \times 22000 = \text{Rp } 742,50$$

$$\text{Kacang almond} = \frac{47,5}{1000} \times 300000 = \text{Rp } 14250,00$$

Tabel 4.4 Tabel Hasil Perhitungan Total Harga

Hari ke-	Waktu Makan	Nama Makanan	Jumlah Berat	Harga
1	Pagi	Kentang	997.5	11970
		Putih telur ayam	146.25	2632.5
		Kacang Tanah	33.75	742.5
		Kacang Almond	47.5	14250
	Siang	Mi Kering	380	7600
		Udang segar	126	7056
		Tahu	396	3960
		Kacang Almond	76	22800
	Malam	Roti Gandum	465.5	20349.3325
		Hati sapi	110.25	6063.75
		Kacang Tolo	63	945
		Kelapa	99.75	798

2	Pagi	Singkong	570	1425
		Cumi-cumi	101.25	4050
		Kembang Tahu	45	7425
		Minyak Kacang Kedele	23.75	2113.75
	Siang	Mi Kering	380	7600
		Rebon kering	36	3204
		Selai Kacang Tanah	54	6912
		Santan	304	10032
	Malam	Nasi Beras Giling	665	6716.5
		Kerang	283.5	9072
		Oncom	126	4410
		Minyak Zaitun	33.25	4987.5
		Jumlah		167114.8325

4.2.4 Perhitungan Penalti Gizi

Perhitungan penalti gizi secara umum adalah selisih antara kandungan kalori dan zat gizi mikro yang dikonsumsi dengan kandungan kalori dan zat gizi mikro yang dibutuhkan. Zat gizi mikro yang dimaksud meliputi karbohidrat, protein, dan lemak. Dalam proses perhitungan penalti gizi hal pertama yang dilakukan adalah menghitung takaran setiap bahan makanan pada masing-masing Partikel.

Takaran bahan makanan untuk masing-masing anggota keluarga diambil dari hasil perkalian antara berat standar yang ada pada daftar bahan makanan penukar dengan porsi yang dianjurkan sesuai dengan standar porsi. Hasil perkalian tersebut kembali dikalikan dengan persentase sesuai waktu makan. Contoh perhitungan pada partikel X1 untuk porsi makan ayah pada hari ke-1 adalah sebagai berikut.

Makan pagi : $Kentang = 210 \times 8 \times 25\% = 420 \text{ gram}$
 $Putih Telur Ayam = 65 \times 3 \times 25\% = 48,75 \text{ gram}$
 $Kacang Tanah = 15 \times 3 \times 25\% = 11.25 \text{ gram}$
 $Kacang Almond = 10 \text{ gram} \times 7 \times 25\% = 17.5 \text{ gram}$

Makan siang : $Mi Kering = 50 \times 8 \times 40\% = 160 \text{ gram}$
 $Udang Segar = 35 \times 3 \times 40\% = 42 \text{ gram}$
 $Tahu = 110 \times 3 \times 40\% = 132 \text{ gram}$
 $Kacang almond = 10 \times 7 \times 40\% = 28 \text{ gram}$

Makan malam : $Roti gandum = 70 \times 8 \times 35\% = 196 \text{ gram}$
 $Hati sapi = 35 \times 3 \times 35\% = 36,75 \text{ gram}$
 $Kacang tolo = 20 \times 3 \times 35\% = 21 \text{ gram}$
 $Kelapa = 15 \times 7 \times 35\% = 36,75 \text{ gram}$

Tabel 4. 5 Tabel Hasil Perhitungan Jumlah Berat Bahan Makanan

Hari ke-	Waktu Makan	Nama Makanan	Berat			Jumlah Berat
			Ayah	Ibu	Anak 1	
1	Pagi	Kentang	420	262.5	315	997.5
		Putih telur ayam	48.75	48.75	48.75	146.25
		Kacang Tanah	11.25	11.25	11.25	33.75
		Kacang Almond	17.5	15	15	47.5
	Siang	Mi Kering	160	100	120	380
		Udang segar	42	42	42	126
		Tahu	132	132	132	396
		Kacang Almond	28	24	24	76
	Malam	Roti Gandum	196	122.5	147	465.5
		Hati sapi	36.75	36.75	36.75	110.25
		Kacang Tolo	21	21	21	63
		Kelapa	36.75	31.5	31.5	99.75
2	Pagi	Singkong	240	150	180	570
		Cumi-cumi	33.75	33.75	33.75	101.25
		Kembang Tahu	15	15	15	45
		Minyak Kacang Kedele	8.75	7.5	7.5	23.75
	Siang	Mi Kering	160	100	120	380
		Rebon kering	12	12	12	36
		Selai Kacang Tanah	18	18	18	54
		Santan	112	96	96	304
	Malam	Nasi Beras Giling	280	175	210	665
		Kerang	94.5	94.5	94.5	283.5
		Oncom	42	42	42	126
		Minyak Zaitun	12.25	10.5	10.5	33.25

Langkah selanjutnya adalah menghitung kandungan kalori dari bahan makanan sesuai takaran. Pada daftar bahan makanan penukar juga telah tersedia kandungan gizi sesuai dengan berat standar. Contoh perhitungan kandungan kalori pada partikel X1 untuk porsi makan pagi ayah hari ke-1 adalah sebagai berikut.

$$\text{Kentang} = 420 \div 210 \times 175 = 350 \text{ kkal}$$

$$\text{Putih Telur Ayam} = 48,75 \div 65 \times 50 = 37,5 \text{ kkal}$$

$$\text{Kacang tanah} = 11,25 \div 15 \times 75 = 56,25 \text{ kkal}$$

$$\text{Kacang almond} = 17.5 \div 10 \times 50 = 87,5 \text{ kkal}$$

Tabel 4.6 Tabel Hasil Perhitungan Kandungan Kalori

Hari ke-	Waktu Makan	Nama Makanan	Kalori		
			Ayah	Ibu	Anak 1
1	Pagi	Kentang	350	218.75	262.5

2		Putih telur ayam	37.5	37.5	37.5
		Kacang Tanah	56.25	56.25	56.25
		Kacang Almond	87.5	75	75
	Siang	Mi Kering	560	350	420
		Udang segar	60	60	60
		Tahu	90	90	90
		Kacang Almond	140	120	120
	Malam	Roti Gandum	490	306.25	367.5
		Hati sapi	78.75	78.75	78.75
		Kacang Tolo	78.75	78.75	78.75
		Kelapa	122.5	105	105
	Pagi	Singkong	350	218.75	262.5
		Cumi-cumi	37.5	37.5	37.5
		Kembang Tahu	56.25	56.25	56.25
		Minyak Kacang Kedele	87.5	75	75
	Siang	Mi Kering	560	350	420
		Rebon kering	60	60	60
		Selai Kacang Tanah	90	90	90
		Santan	140	120	120
	Malam	Nasi Beras Giling	490	306.25	367.5
		Kerang	52.5	52.5	52.5
		Oncom	78.75	78.75	78.75
		Minyak Zaitun	122.5	105	105
		Jumlah	4276.25	3126.25	3476.25

Lalu setelah itu menghitung kandungan karbohidrat dari setiap bahan makanan yang dikonsumsi. Contoh perhitungan kandungan karbohidrat pada partikel X1 untuk porsi makan pagi ayah pada hari ke-1 adalah sebagai berikut.

$$\text{Kentang} = 420 \div 210 \times 40 = 80 \text{ gram}$$

$$\text{Putih Telur Ayam} = 48,75 \div 65 \times 0 = 0 \text{ gram}$$

$$\text{Kacang tanah} = 11,25 \div 15 \times 7 = 5,25 \text{ gram}$$

$$\text{Kacang almond} = 17,5 \div 10 \times 0 = 0 \text{ gram}$$

Tabel 4.7 Tabel Hasil Perhitungan Kandungan Karbohidrat

Hari ke-	Waktu Makan	Nama Makanan	Karbohidrat		
			Ayah	Ibu	Anak 1
1	Pagi	Kentang	80	50	60
		Putih telur ayam	0	0	0
		Kacang Tanah	5.25	5.25	5.25
		Kacang Almond	0	0	0
	Siang	Mi Kering	128	80	96
		Udang segar	0	0	0

	Malam	Tahu	8.4	8.4	8.4
		Kacang Almond	0	0	0
		Roti Gandum	112	70	84
		Hati sapi	0	0	0
		Kacang Tolo	7.35	7.35	7.35
		Kelapa	0	0	0
2	Pagi	Singkong	80	50	60
		Cumi-cumi	0	0	0
		Kembang Tahu	5.25	5.25	5.25
		Minyak Kacang Kedele	0	0	0
	Siang	Mi Kering	128	80	96
		Rebon kering	0	0	0
		Selai Kacang Tanah	8.4	8.4	8.4
		Santan	0	0	0
	Malam	Nasi Beras Giling	112	70	84
		Kerang	0	0	0
		Oncom	7.35	7.35	7.35
		Minyak Zaitun	0	0	0
		Jumlah	682	442	522

Langkah berikutnya yaitu menghitung kandungan protein setiap bahan makanan yang dikonsumsi sesuai dengan takaran. Contoh perhitungan kandungan protein pada partikel X1 untuk porsi makan pagi ayah pada hari ke-1 adalah sebagai berikut.

$$\text{Kentang} = 420 \div 210 \times 4 = 8 \text{ gram}$$

$$\text{Putih Telur Ayam} = 48,75 \div 65 \times 7 = 5,25 \text{ gram}$$

$$\text{Kacang tanah} = 11,25 \div 15 \times 5 = 53,75 \text{ gram}$$

$$\text{Kacang almond} = 17.5 \div 10 \times 0 = 0 \text{ gram}$$

Tabel 4.8 Tabel Hasil Perhitungan Kandungan Protein

Hari ke-	Waktu Makan	Nama Makanan	Protein		
			Ayah	Ibu	Anak 1
1	Pagi	Kentang	8	5	6
		Putih telur ayam	5.25	5.25	5.25
		Kacang Tanah	3.75	3.75	3.75
		Kacang Almond	0	0	0
	Siang	Mi Kering	12.8	8	9.6
		Udang segar	8.4	8.4	8.4
		Tahu	6	6	6
		Kacang Almond	0	0	0
	Malam	Roti Gandum	11.2	7	8.4
		Hati sapi	7.35	7.35	7.35

2		Kacang Tolo	5.25	5.25	5.25
		Kelapa	0	0	0
	Pagi	Singkong	8	5	6
		Cumi-cumi	5.25	5.25	5.25
		Kembang Tahu	3.75	3.75	3.75
		Minyak Kacang Kedele	0	0	0
	Siang	Mi Kering	12.8	8	9.6
		Rebon kering	8.4	8.4	8.4
		Selai Kacang Tanah	6	6	6
		Santan	0	0	0
	Malam	Nasi Beras Giling	11.2	7	8.4
		Kerang	7.35	7.35	7.35
		Oncom	5.25	5.25	5.25
		Minyak Zaitun	0	0	0
		Jumlah	136	112	120

Tahapan selanjutnya menghitung kandungan lemak setiap bahan makanan yang dikonsumsi sesuai dengan takaran. Contoh perhitungan kandungan lemak pada partikel X1 untuk porsi makan pagi ayah pada hari ke-1 adalah sebagai berikut.

$$\text{Kentang} = 420 \div 21004 = 0.01 \text{ gram}$$

$$\text{Putih Telur Ayam} = 48,75 \div 65 \times 2 = 1,5 \text{ gram}$$

$$\text{Kacang tanah} = 11,25 \div 15 \times 3 = 2,25 \text{ gram}$$

$$\text{Kacang almond} = 17.5 \div 10 \times 5 = 8,75 \text{ gram}$$

Tabel 4.9 Tabel Hasil Perhitungan Kandungan Lemak

Hari ke-	Waktu Makan	Nama Makanan	Lemak		
			Ayah	Ibu	Anak 1
1	Pagi	Kentang	0	0	0
		Putih telur ayam	1.5	1.5	1.5
		Kacang Tanah	2.25	2.25	2.25
		Kacang Almond	8.75	7.5	7.5
	Siang	Mi Kering	0	0	0
		Udang segar	2.4	2.4	2.4
		Tahu	3.6	3.6	3.6
		Kacang Almond	14	12	12
	Malam	Roti Gandum	0	0	0
		Hati sapi	5.25	5.25	5.25
		Kacang Tolo	3.15	3.15	3.15
		Kelapa	12.25	10.5	10.5
2	Pagi	Singkong	0	0	0
		Cumi-cumi	1.5	1.5	1.5

		Kembang Tahu	2.25	2.25	2.25
		Minyak Kacang Kedele	8.75	7.5	7.5
	Siang	Mi Kering	0	0	0
		Rebon kering	2.4	2.4	2.4
		Selai Kacang Tanah	3.6	3.6	3.6
		Santan	14	12	12
		Nasi Beras Giling	0	0	0
	Malam	Kerang	2.1	2.1	2.1
		Oncom	3.15	3.15	3.15
		Minyak Zaitun	12.25	10.5	10.5
		Jumlah	103.15	93.15	93.15

Setelah semua kandungan zat gizi dihitung, selanjutnya dilakukan perhitungan penalti gizi.

$$\text{Penalti Gizi} = \sum (\text{Keb. energi} - \text{Kandungan kalori}) + (\text{Keb. karbohidrat} - \text{Kandungan karbohidrat}) + (\text{Keb. protein} - \text{Kandungan protein}) + (\text{Keb. Lemak} - \text{Kandungan Lemak}) \quad (4.3)$$

Contoh perhitungan penalti gizi diambil dari partikel X1.

$$\begin{aligned} \text{Penalti Gizi} = & ((4276,25 - 2745 \div 2) + (682 - 377 \div 2) \\ & + (136 - 102,92 \div 2) + (103,15 - 91,40 \div 2) \\ & + ((3126,25 - 2178 \div 2) + (442 - 299 \div 2) \\ & + (112 - 81,65 \div 2) + (93,15 - 72,58 \div 2)) \\ & + ((3476,25 - 2278 \div 2) + (522 - 330 \div 2) \\ & + (120 - 68,33 \div 2) + (93,15 - 75,92 \div 2)) = 2108,86 \end{aligned}$$

Perhitungan Variasi

Tujuan dari penghitungan variasi adalah untuk mengetahui banyaknya variasi dari bahan makanan. Jika tidak terjadi duplikasi atau kesamaan bahan makanan dalam satu partikel maka variasi bernilai 1. Namun jika terjadi duplikasi atau kesamaan untuk bahan makanan tersebut dinilai 0. Dalam contoh perhitungan manual ini, variasi untuk Partikel X1 bernilai 20.

4.2.5 Perhitungan *Fitness*

Perhitungan *fitness* dilakukan untuk semua partikel pada setiap iterasi. Perhitungan *fitness* menjadi acuan untuk menentukan *Pbest* dan *Gbest* setiap kali iterasi. Mengacu pada Persamaan 2.4 dan 2.5 persamaan *fitness* untuk kasus ini adalah:

$$f(x) = \frac{1}{\text{Penalti Gizi}} \cdot C1 + \frac{1}{\text{Harga Total}} \cdot C2 + \text{Variasi} \quad (4.4)$$

Keterangan :

C1 & C2 : bilangan untuk menyeimbangkan nilai *fitness*

Penggunaan konstanta $C1$ dan $C2$ pada persamaan di atas bertujuan agar nilai *fitness* yang dihasilkan bisa seimbang. Jika nilai yang dihasilkan dalam perhitungan variasi berupa angka puluhan, maka akan terjadi ketidakseimbangan jika pembagian penalti gizi dan harga total menghasilkan angka berupa angka desimal. Konstanta $C1$ untuk pembagian penalti gizi yaitu 100000, sedangkan konstanta $C2$ untuk pembagian harga total yaitu 10000000. Sebagai contoh perhitungan nilai *fitness* partikel X1 adalah sebagai berikut.

$$f(X1) = \frac{1}{167114,83} \cdot 100000 + \frac{1}{2108,86} \cdot 10000000 + 20$$

$$= 59,83 + 47,41 + 20 = 127.248$$

4.2.6 Penentuan Pbest

Nilai *Pbest* didapat dengan membandingkan nilai *fitness* setiap partikel pada setiap iterasi, artinya akan dilakukan perbandingan *fitness* antara partikel ke- i pada iterasi ke- j dengan *fitness value* Partikel ke- i dengan iterasi ke- $(j-1)$, Partikel yang menghasilkan *fitness* paling baik dalam proses perbandingan tersebut akan dijadikan *Pbest*.

Berikut adalah hasil *fitness* dan *Pbest* dari partikel X1 dan X2 pada iterasi 0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Fitness
X1	11	24	19	6	23	31	50	7	42	39	26	39	46	3	34	20	24	26	45	35	28	23	39	25	127.2580
X2	29	15	21	9	17	55	31	7	3	39	31	50	46	16	4	34	37	41	12	27	35	3	32	12	127.5009

Berikut adalah hasil *fitness* dan *Pbest* dari partikel X1 dan X2 pada iterasi 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Fitness
X1	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15	134.5832
X2	29	15	21	9	17	55	31	7	3	39	31	50	46	16	4	34	37	41	12	27	35	3	32	12	127.5009

Berikut adalah hasil *fitness* dan *Pbest* dari partikel X1 dan X2 pada iterasi 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Fitness
X1	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15	134.5832
X2	22	19	20	8	19	45	39	7	18	39	29	46	46	11	16	28	32	35	25	30	32	11	35	17	134.5832

4.2.7 Penentuan Gbest

Nilai *Gbest* diambil dari partikel dengan nilai *fitness* terbaik pada sebuah swarm atau populasi.

Berikut adalah *Gbest* dari iterasi 0.

X1	29	15	21	9	17	55	31	7	3	39	31	50	46	16	4	34	37	41	12	27	35	3	32	12	127.5009
----	----	----	----	---	----	----	----	---	---	----	----	----	----	----	---	----	----	----	----	----	----	---	----	----	----------

Berikut adalah *Gbest* dari iterasi 1.

X2	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15	134.5832
----	----	----	----	---	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----------

Berikut adalah *Gbest* dari iterasi 2.

X2	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15	134.5832
----	----	----	----	---	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----------

4.2.8 Perhitungan Perubahan Kecepatan Partikel

Untuk menentukan perubahan kecepatan partikel, dibutuhkan perhitungan nilai inersia (ω) yang bisa didapat dengan persamaan berikut seperti sudah disebutkan pada Persamaan 2.2.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{IterMax} * iter$$

Dengan parameter yang sudah ditentukan sebelumnya yaitu:

$$\omega_{min} = 0.4$$

$$\omega_{max} = 0.9$$

$$C_1 = 2$$

$$C_2 = 2$$

$$IterMax = 2$$

Perhitungan inersia untuk iterasi 1 adalah:

$$\omega = 0.9 - \frac{0.9 - 0.4}{2} * 0 = 0.9$$

Lalu yang diperlukan lagi untuk perhitungan kecepatan penentuan nilai random untuk R1 dan R2. Pada perhitungan manual ini, ditentukan nilai random R1 dan R2 untuk iterasi 1 adalah R1=0.934139 dan R2 = 0.396350.

Sedangkan untuk perhitungan perubahan kecepatan sendiri, digunakan persamaan seperti Persamaan 2-1 yaitu.

$$V_{i,j}^k = \omega \cdot V_{i,j}^{k-1} + C_1 R_1 (Pbest_{i,j}^{k-0} - X_{i,j}^{k-1}) + C_2 R_2 (Gbest_{i,j}^{k-0} - X_{i,j}^{k-1})$$

Sebagai contoh perhitungan untuk kecepatan partikel X1 adalah:

$$V_{1,1}^1 = 0.9 * 0 + 2 * 0.934139(11 - 11) + 2 * 0.396350(29 - 11)$$

$$V_{1,1}^1 = 14,27$$

Pada Tabel berikut adalah perubahan kecepatan partikel pada untuk iterasi 1:

Tabel 4.10 Tabel Contoh Perubahan Kecepatan Partikel

	1	2	3	4	5	6	7	8	9	10	11	12
V1	14.27	-7.13	1.59	2.38	-4.76	19.02	-15.06	0.00	-30.92	0.00	3.96	8.72
	13	14	15	16	17	18	19	20	21	22	23	24
	0.00	10.31	-23.78	11.10	10.31	11.89	-26.16	-6.34	5.55	-15.85	-5.55	-10.31
	1	2	3	4	5	6	7	8	9	10	11	12
V2	0	0	0	0	0	0	0	0	0	0	0	0

	13	14	15	16	17	18	19	20	21	22	23	24
	0	0	0	0	0	0	0	0	0	0	0	0

4.2.9 Perhitungan Perubahan Posisi Partikel

Berhubungan dengan perubahan kecepatan partikel, posisi partikel pun juga ikut berubah. Perubahan posisi partikel dapat dihitung dengan persamaan berikut:

$$X_{i,j}^k = X_{i,j}^{k-1} + V_{i,j}^k \quad (4.6)$$

Sebagai contoh adalah perhitungan posisi untuk X1 iterasi 1 berikut:

$$X_{1,1}^1 = X_{1,1}^0 + V_{1,1}^1$$

$$X_{1,1}^1 = 11 + 14,27$$

$$X_{1,1}^1 = 24,27 \sim 25$$

Tabel Berikut adalah hasil perhitungan perubahan posisi partikel untuk iterasi 1 secara keseluruhan:

Tabel 4.11 Tabel Contoh Perubahan Posisi Partikel

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
X1	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15
X2	29	15	21	9	17	55	31	7	3	39	31	50	46	16	4	34	37	41	12	27	35	3	32	12

4.2.10 Hasil Perhitungan Manual

Dari semua tahapan perhitungan manual yang sudah dilakukan, maka Optimasi dengan menggunakan algoritma *Particle swarm optimization* dengan parameter $\omega_{min} = 0.4$, $\omega_{max} = 0.9$, $C_1 = 2$, $C_2 = 2$ dan $IterMax = 2$ adalah :

X2	25	17	21	8	18	50	35	7	11	39	30	48	46	13	10	31	34	38	19	29	34	7	33	15	134.5832
----	----	----	----	---	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----------

Dengan penjabaran sebagai berikut :

Tabel 4.12 Data Bahan Makanan Hasil Perhitungan Manual

Hari ke-	Waktu Makan	Jenis Makanan	Bahan	Gen	Indeks Makanan	Nama Makanan
1	Pagi	KH		25	8	Mi Kering
		PH		17	12	Ikan mujair
		PN		21	5	Kacang Tanah
		L		8	3	Mayonnaise
	Siang	KH		18	6	Makaroni
		PH		50	36	Corned beef
		PN		35	9	Oncom
		L		7	2	Kacang Almond
	Malam	KH		11	4	Kentang
		PH		39	28	Hati sapi

2		PN	30	7	Kacang Tolo
		L	48	11	Minyak Kelapa
	Pagi	KH	46	14	Singkong
		PH	13	10	Ikan lele
		PN	10	3	Kacang Merah
		L	31	7	Mentega
	Siang	KH	34	11	Nasi Ketan Putih
		PH	38	28	Hati sapi
		PN	19	5	Kacang Tanah
		L	29	7	Mentega
	Malam	KH	34	11	Nasi Ketan Putih
		PH	7	5	Dendeng daging sapi
		PN	33	8	Kembang Tahu
		L	15	4	Minyak Jagung

Tabel 4.13 Harga dan Berat Makanan Hasil Perhitungan Manual

Hari ke-	Waktu Makan	Indeks Bahan Makanan	Nama Makanan	Berat			Jumlah Berat	Harga
				Ayah	Ibu	Anak 1		
1	Pagi	8	Mi Kering	100	62.5	75	237.5	4750.00
		12	Ikan mujair	22.5	22.5	22.5	67.5	2025.00
		5	Kacang Tanah	11.25	11.25	11.25	33.75	742.50
		3	Mayonnaise	35	30	30	95	2850.00
	Siang	6	Makaroni	160	100	120	380	11400.00
		36	Corned beef	54	54	54	162	12393.00
		9	Oncom	48	48	48	144	5040.00
		2	Kacang Almond	28	24	24	76	22800.00
	Malam	4	Kentang	588	367.5	441	1396.5	16758.00
		28	Hati sapi	36.75	36.75	36.75	110.25	6063.75
		7	Kacang Tolo	21	21	21	63	945.00
		11	Minyak Kelapa	12.25	10.5	10.5	33.25	891.10
2	Pagi	14	Singkong	240	150	180	570	1425.00
		10	Ikan lele	30	30	30	90	1800.00
		3	Kacang Merah	15	15	15	45	1080.00
		7	Mentega	8.75	7.5	7.5	23.75	1187.50
	Siang	11	Nasi Ketan Putih	320	200	240	760	10640.00
		28	Hati sapi	42	42	42	126	6930.00
		5	Kacang Tanah	18	18	18	54	1188.00
		7	Mentega	14	12	12	38	1900.00
	Malam	11	Nasi Ketan Putih	280	175	210	665	9310.00
		5	Dendeng daging sapi	15.75	15.75	15.75	47.25	18427.50
		8	Kembang Tahu	21	21	21	63	10395.00

		4	Minyak Jagung	12.25	10.5	10.5	33.25	3424.75
			Jumlah Harga					154366.10

Tabel 4.14 Kandungan Kalori Bahan Makanan Hasil Perhitungan Manual

Hari ke-	Waktu Makan	Nama Makanan	Kalori		
			Ayah	Ibu	Anak 1
1	Pagi	Mi Kering	350	218.75	262.5
		Ikan mujair	37.5	37.5	37.5
		Kacang Tanah	56.25	56.25	56.25
		Mayonnaise	87.5	75	75
	Siang	Makaroni	560	350	420
		Corned beef	180	180	180
		Oncom	90	90	90
		Kacang Almond	140	120	120
	Malam	Kentang	490	306.25	367.5
		Hati sapi	78.75	78.75	78.75
		Kacang Tolo	78.75	78.75	78.75
		Minyak Kelapa	122.5	105	105
2	Pagi	Singkong	350	218.75	262.5
		Ikan lele	37.5	37.5	37.5
		Kacang Merah	56.25	56.25	56.25
		Mentega	87.5	75	75
	Siang	Nasi Ketan Putih	560	350	420
		Hati sapi	90	90	90
		Kacang Tanah	90	90	90
		Mentega	140	120	120
	Malam	Nasi Ketan Putih	490	306.25	367.5
		Dendeng daging sapi	52.5	52.5	52.5
		Kembang Tahu	78.75	78.75	78.75
		Minyak Jagung	122.5	105	105
		Jumlah Harga	4426.25	3276.25	3626.25

Tabel 4.15 Kandungan Karbohidrat, Protein dan Lemak Bahan Makanan Hasil Perhitungan Manual

Ha ri ke-	Waktu Makan	Nama Makanan	Karbohidrat			Protein			Lemak		
			Ayah	Ibu	Anak 1	Ayah	Ibu	Anak 1	Ayah	Ibu	Anak 1
1	Pagi	Mi Kering	80	50	60	8	5	6	0	0	0
		Ikan mujair	0	0	0	5.25	5.25	5.25	1.5	1.5	1.5
		Kacang Tanah	5.25	5.25	5.25	3.75	3.75	3.75	2.25	2.25	2.25
		Mayonnaise	0	0	0	0	0	0	8.75	7.5	7.5
	Siang	Makaroni	128	80	96	12.8	8	9.6	0	0	0
		Corned beef	0	0	0	8.4	8.4	8.4	15.6	15.6	15.6
		Oncom	8.4	8.4	8.4	6	6	6	3.6	3.6	3.6
		Kacang Almond	0	0	0	0	0	0	14	12	12
	Malam	Kentang	112	70	84	11.2	7	8.4	0	0	0
		Hati sapi	0	0	0	7.35	7.35	7.35	5.25	5.25	5.25
		Kacang Tolo	7.35	7.35	7.35	5.25	5.25	5.25	3.15	3.15	3.15
		Minyak Kelapa	0	0	0	0	0	0	12.25	10.5	10.5
2	Pagi	Singkong	80	50	60	8	5	6	0	0	0
		Ikan lele	0	0	0	5.25	5.25	5.25	1.5	1.5	1.5
		Kacang Merah	5.25	5.25	5.25	3.75	3.75	3.75	2.25	2.25	2.25
		Mentega	0	0	0	0	0	0	8.75	7.5	7.5
	Siang	Nasi Ketan Putih	128	80	96	12.8	8	9.6	0	0	0
		Hati sapi	0	0	0	8.4	8.4	8.4	6	6	6
		Kacang Tanah	8.4	8.4	8.4	6	6	6	3.6	3.6	3.6
		Mentega	0	0	0	0	0	0	14	12	12
	Malam	Nasi Ketan Putih	112	70	84	11.2	7	8.4	0	0	0
		Dendeng daging sapi	0	0	0	7.35	7.35	7.35	2.1	2.1	2.1
		Kembang Tahu	7.35	7.35	7.35	5.25	5.25	5.25	3.15	3.15	3.15
		Minyak Jagung	0	0	0	0	0	0	12.25	10.5	10.5
		Jumlah Harga	682.0	442.0	522.0	136.0	112.0	120.0	119.9	109.9	109.9
			0	0	0	0	0	0	5	5	5

4.3 Perancangan Antarmuka

Antarmuka digunakan sebagai fasilitas interaksi antara pengguna dengan sistem. Perancangan antarmuka berfungsi untuk menggambarkan antarmuka sistem yang akan diterapkan pada tahap implementasi. Antarmuka sistem optimasi bahan makanan menggunakan algoritma *Particle swarm optimization* dibagi menjadi 3 bagian, yaitu data keluarga, data bahan makanan, dan optimasi menggunakan algoritma *Particle swarm optimization*.

Untuk halaman utama pada antarmuka ini terdiri dari 4 panel yaitu panel parameter *particle swarm optimization*, panel *input* data keluarga, panel info dan

Gambar 4.14 Rancangan Antarmuka Halaman Utama

Untuk penjelasan rancangan antarmuka halaman utama yang ditunjukkan pada Gambar 4.14 adalah sebagai berikut:

- 54

5. Panel hasil input data keluarga.
6. Panel hasil kebutuhan gizi keluarga.
7. Panel hasil optimasi.
8. Panel keterangan bahan makanan.
9. *Textfield* untuk memasukkan parameter ω_{min} .
10. *Textfield* untuk memasukkan parameter ω_{max} .
11. *Textfield* untuk memasukkan parameter C_1 .
12. *Textfield* untuk memasukkan parameter C_2 .
13. *Textfield* untuk memasukkan parameter iterasi maksimal.
14. *Textfield* untuk memasukkan parameter jumlah partikel.
15. *Textfield* untuk memasukkan parameter jumlah 1 partikel.
16. *Textfield* untuk memasukkan parameter jumlah hari.
17. *Button default* untuk mengeset nilai parameter menjadi nilai *default*.
18. *Button refresh* untuk menghapus nilai parameter.
19. *Button Proses* untuk menjalankan proses optimasi.
20. *Textfield* untuk memasukkan nama anggota keluarga.
21. *Textfield* untuk memasukkan usia anggota keluarga.
22. *Textfield* untuk memasukkan berat badan anggota keluarga.
23. *Textfield* untuk memasukkan tinggi badan anggota keluarga.
24. *Radio button* untuk memilih jenis kelamin anggota keluarga.
25. *Combo box* untuk memilih jenis aktivitas anggota keluarga.
26. *Button default* untuk mengeset data keluarga menjadi *data default*.
27. *Button simpan* untuk menyimpan data keluarga yang diinputkan.
28. *Textfield* untuk menampilkan *running time* sistem.

4.4 Perancangan Uji Coba dan Evaluasi

Perancangan uji coba dan evaluasi disebabkan karena belum adanya metode yang pasti untuk melakukan optimasi pemenuhan gizi keluarga menggunakan algoritma *particle swarm optimization*. Oleh karena itu untuk melakukan proses merancang langkah evaluasi sistem yang telah dikembangkan maka akan dilakukan uji coba antara lain:

1. Uji coba berdasarkan banyaknya partikel
2. Uji coba berdasarkan kombinasi nilai ω_{min} dan ω_{max}
3. Uji coba berdasarkan banyaknya iterasi
4. Uji coba berdasarkan batas angka permutasi

4.4.1 Uji Coba Berdasarkan Banyaknya Partikel

Uji coba berdasarkan banyaknya partikel bertujuan untuk mengetahui hubungan antara jumlah partikel terhadap nilai *fitness*. Banyaknya partikel yang digunakan dalam uji coba ini adalah bilangan kelipatan 5, mulai 5 hingga 50. Uji coba banyaknya partikel dilakukan sebanyak 10 kali dengan parameter lain yang ditentukan.

Tabel 4.16 Tabel Perancangan Uji Coba Berdasarkan Banyaknya Partikel

Jumlah Partikel	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
5											
10											
15											
20											
25											
30											
35											
40											
45											
50											

4.4.2 Uji Coba berdasarkan Banyaknya Iterasi

Uji coba berdasarkan banyaknya iterasi bertujuan untuk mengetahui hubungan antara jumlah iterasi terhadap nilai *fitness*. Nilai yang diujikan adalah bilangan kelipatan 10 mulai 10 hingga 100.

Tabel 4.17 Tabel Perancangan Uji Coba Berdasarkan Banyaknya Iterasi

Iterasi	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											
80											
90											
100											

4.4.3 Uji Coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}

Pengujian kombinasi nilai bobot inersia ω_{min} dan ω_{max} digunakan untuk menentukan kombinasi nilai bobot inersia ω_{min} dan ω_{max} yang optimal yang juga akan menghasilkan solusi terbaik. Pengujian dilakukan 10 kali untuk masing-masing nilai kombinasi bobot inersia. Rentang yang digunakan adalah antara 0.4 sampai dengan 0.9 karena dianggap optimal.

Tabel 4.18 Tabel Perancangan Uji Coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}

$\omega_{min};$ ω_{max}	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
0,4;0,5											
0,4;0,6											
0,4;0,7											
0,4;0,8											
0,4;0,9											

4.4.4 Uji Coba Berdasarkan Batas Angka Permutasi

Uji coba berdasarkan batas angka permutasi bertujuan untuk mengetahui apakah batas atas angka permutasi memiliki pengaruh terhadap nilai *fitness* yang akan dihasilkan. Nilai batas angka permutasi yang diujikan adalah bilangan mulai 55 hingga 145 dengan kelipatan 10.

Tabel 4.19 Tabel Perancangan Uji Coba Berdasarkan Batas Angka Permutasi

Range	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
55											
65											
75											
85											
95											
105											
115											
125											
135											
145											

BAB 5 IMPLEMENTASI

Pada bagian ini akan dijelaskan mengenai implementasi sistem yang berdasar pada bab sebelumnya yaitu bab perancangan, yang meliputi perancangan algoritma dan perancangan antarmuka.

5.1 Implementasi Kode Program

Berdasarkan bab sebelumnya yaitu bab perancangan, maka pada bagian ini akan membahas tentang implementasi kode program sesuai dengan perancangan algoritma yang telah dibuat sebelumnya. Sistem akan diimplementasikan menggunakan bahasa pemrograman Java dengan aplikasi Netbeans versi 7.3.1.

5.1.1 Inisialisasi Partikel Awal

Proses inisialisasi partikel awal adalah proses untuk menyiapkan posisi awal partikel, dan kecepatan awal dari partikel yang digunakan. Rentang nilai partikel berkisar dari 1 sampai variabel *BatasAtas* yang diambil secara acak. Struktur penulisan kode program untuk proses inisialisasi partikel awal ditunjukkan pada Kode Program 5.1.

```

1 public int[][] Inisialisasi(int Swarmsize, int Jumlahgen, int BatasAtas){
2     int[][] Swarm = new int[Swarmsize][Jumlahgen];
3     Random random = new Random();
4     for (int i = 0; i < Swarmsize; i++) {
5         for (int j = 0; j < Jumlahgen; j++) {
6             Swarm[i][j] = random.nextInt(BatasAtas) + 1;
7         }
8     }
9     return Swarm;
10
11
12 }
```

Kode Program 5.1 Inisialisasi Partikel Awal

Berikut adalah penjelasan untuk Kode Program 5.1:

1. Baris 1 merupakan deklarasi *method* dan parameternya.
2. Baris 2-10 adalah proses pengisian nilai masing-masing partikel dengan nilai *random* diambil dari bilangan 1 sampai dengan variabel *BatasAtas* yang ditentukan.

5.1.2 Inisialisasi Kecepatan Awal

Proses inisialisasi kecepatan awal adalah proses untuk menyiapkan kecepatan awal dari partikel yang digunakan. Nilai partikel kecepatan partikel awal diset 0 yang nantinya akan berubah seiring dengan bertambahnya iterasi. Struktur

penulisan kode program untuk proses inisialisasi kecepatan awal ditunjukkan pada Kode Program 5.2.

```

1 public int[][] Velocity(int Swarmsize, int Jumlahgen, int velocity) {
2     int[][] Vel = new int[Swarmsize][Jumlahgen];
3     //Random random = new Random();
4     for (int i = 0; i < Swarmsize; i++) {
5         for (int j = 0; j < Jumlahgen; j++) {
6             Vel[i][j] = 0;
7         }
8     }
9     return Vel;
10
11
12 }

```

Kode Program 5.2 Inisialisasi Kecepatan Awal

Berikut adalah penjelasan untuk Kode Program 5.2:

1. Baris 1 adalah deklarasi *method* dan parameternya.
2. Baris 2-10 adalah proses pengisian nilai masing-masing kecepatan partikel dengan nilai awal 0.

5.1.3 Perhitungan *Fitness*

Proses menghitung *fitness* adalah proses untuk menghitung nilai *fitness* yang dihasilkan dari posisi terbaru partikel yang nantinya akan menjadi kandidat solusi optimal. Struktur penulisan kode program untuk proses menghitung nilai *fitness* ditunjukkan pada Kode Program 5.3.

```

1 public double HitungFitness(int[] Partikel, int[] Type, String[] jk,
2 double[] Energi, double[] Karbohidrat, double[] Protein, double[]
3 Lemak) throws SQLException {
4     int[] IndeksBM = this.KonversiIndeksBM(Partikel);
5     ///mengkonversi angka permutasi menjadi indeks bahan makanan
6     String[] NamaBM =
7     this.KonversiNamaBM(IndeksBM); ///mengkonversi indeks bahan makanan
8     menjadi nama bahan makanan
9     double[][] Berat = this.HitungBeratBMPerOrang(IndeksBM, Type,
10 jk); ///menghitung berat masing-masing bahan makanan untuk setiap orang
11     double[] BeratTotal =
12     this.HitungBeratTotal(Berat); ///menghitung berat total setiap bahan
13     makanan untuk satu keluarga
14     double[][] KandunganKalori =
15     this.HitungKaloriPerBMPerOrang(IndeksBM, Berat); ///menghitung
16     kandungan kalori setiap bahan makanan untuk setiap orang
17     double SelisihKaloriTotal =
18     this.HitungSelisihKaloriTotal(Energi, KandunganKalori); ///menghitung
19     selisih kalori yang dikonsumsi dengan kalori yang dibutuhkan
20     double[][] KandunganKarbohidrat =
21     this.HitungKarbohidratPerBMPerOrang(IndeksBM, Berat); ///menghitung
22     kandungan karbohidrat setiap bahan makanan untuk setiap orang
23     double SelisihKarbohidratTotal =
24     this.HitungSelisihKarbohidratTotal(Karbohidrat,
25     KandunganKarbohidrat); ///menghitung selisih karbohidrat yang
26     dikonsumsi dengan karbohidrat yang dibutuhkan
27     double[][] KandunganProtein =

```

```

28 this.HitungProteinPerBMPerOrang(IndeksBM, Berat);//menghitung
29 kandungan protein setiap bahan makanan untuk setiap orang
30 double SelisihProteinTotal =
31 this.HitungSelisiProteinTotal(Protein, KandunganProtein);//menghitung
32 selisih protein yang dikonsumsi dengan protein yang dibutuhkan
33 double[][] KandunganLemak =
34 this.HitungLemakPerBMPerOrang(IndeksBM, Berat);//menghitung kandungan
35 lemak setiap bahan makanan untuk setiap orang
36 double SelisihLemakTotal =
37 this.HitungSelisihLemakTotal(Lemak, KandunganLemak);//menghitung
38 selisih lemak yang dikonsumsi dengan lemak yang dibutuhkan
39 double[] Harga = this.HitungHargaBM(IndeksBM,
40 BeratTotal);//menghitung harga setiap bahan makanan sesuai dengan
41 berat total untuk satu keluarga
42 double HargaTotal =
43 this.HitungHargaTotalBM(Harga);//menghitung harga total bahan makanan
44 int Variasi = this.HitungVariasi>NamaBM);//menghitung variasi
45 bahan makanan
46 //menghitung nilai fitness
47 double Fitness = (100000 / (SelisihKaloriTotal +
48 SelisihKarbohidratTotal
49 + SelisihProteinTotal + SelisihLemakTotal)
50 + 10000000 / HargaTotal + Variasi);
51 return Fitness;
    }

```

Kode Program 5.3 Perhitungan *Fitness*

Berikut adalah penjelasan untuk Kode Program 5.3:

1. Baris 1 merupakan deklarasi *method* dan parameternya.
2. Baris 4-5 adalah proses mengkonversi angka permutasi menjadi indeks bahan makanan.
3. Baris 6-7 adalah proses mengkonversi indeks bahan makanan menjadi nama bahan makanan
4. Baris 8-9 adalah proses menghitung berat masing-masing bahan makanan untuk setiap orang
5. Baris 10-11 adalah proses menghitung berat total setiap bahan makanan untuk satu keluarga
6. Baris 13-14 adalah proses menghitung kandungan kalori setiap bahan makanan untuk setiap orang
7. Baris 16-17 adalah proses menghitung selisih kalori yang dikonsumsi dengan kalori yang dibutuhkan
8. Baris 19-20 adalah proses menghitung kandungan karbohidrat setiap bahan makanan untuk setiap orang
9. Baris 22-23 adalah proses menghitung selisih karbohidrat yang dikonsumsi dengan karbohidrat yang dibutuhkan
10. Baris 26-27 adalah proses menghitung kandungan protein setiap bahan makanan untuk setiap orang
11. Baris 29-30 adalah proses menghitung selisih protein yang dikonsumsi dengan protein yang dibutuhkan
12. Baris 32-33 adalah proses menghitung kandungan lemak setiap bahan makanan untuk setiap orang
13. Baris 36 adalah proses menghitung selisih lemak yang dikonsumsi dengan lemak yang dibutuhkan

14. Baris 38-39 adalah proses menghitung harga setiap bahan makanan sesuai dengan berat total untuk satu keluarga
15. Baris 41-42 adalah proses menghitung harga total bahan makanan
16. Baris 43 adalah proses menghitung variasi bahan makanan
17. Baris 46-50 adalah proses menghitung nilai *fitness*

5.1.3.2 Konversi Indeks Bahan Makanan

Proses konversi indeks bahan makanan adalah bagian dari perhitungan *fitness* dimana angka permutasi *random* partikel dikonversi menjadi indeks bahan makanan. Struktur penulisan kode program untuk proses konversi indeks bahan makanan ditunjukkan pada Kode Program 5.4.

```

1. public int[] KonversiIndeksBM(int[] Partikel) throws SQLException {
2.     Koneksi koneksi = new Koneksi();
3.     Connection conn = koneksi.getKoneksi();
4.     Statement statement = conn.createStatement();
5.     int Length = Partikel.length;
6.     ResultSet rs;
7.     int[] IndeksBM = new int[Length];
8.     for (int j = 0; j < Length; j = j + 9) {
9.         IndeksBM[j] = (int) Math.round(((this.CountRow(conn,
10. "karbohidrt") - 1) * ((double) (Partikel[j] - 1) / (55 - 1))) + 1);
11.         IndeksBM[j + 1] = (int) Math.round(((this.CountRow(conn,
12. "protein_hewani") - 1) * ((double) (Partikel[j + 1] - 1) / (55 -
13. 1)) + 1);
14.         IndeksBM[j + 2] = (int) Math.round(((this.CountRow(conn,
15. "protein_nabati") - 1) * ((double) (Partikel[j + 2] - 1) / (55 -
16. 1)) + 1);
17.         IndeksBM[j + 3] = (int) Math.round(((this.CountRow(conn,
18. "sayurana") - 1) * ((double) (Partikel[j + 3] - 1) / (55 - 1)) +
19. 1);
20.         IndeksBM[j + 4] = (int) Math.round(((this.CountRow(conn,
21. "sayuranb") - 1) * ((double) (Partikel[j + 4] - 1) / (55 - 1)) +
22. 1);
23.         IndeksBM[j + 5] = (int) Math.round(((this.CountRow(conn,
24. "buah") - 1) * ((double) (Partikel[j + 5] - 1) / (55 - 1)) + 1);
25.         IndeksBM[j + 6] = (int) Math.round(((this.CountRow(conn,
26. "lemak") - 1) * ((double) (Partikel[j + 6] - 1) / (55 - 1)) + 1);
27.         IndeksBM[j + 7] = (int) Math.round(((this.CountRow(conn,
28. "gula") - 1) * ((double) (Partikel[j + 7] - 1) / (55 - 1)) + 1);
29.         IndeksBM[j + 8] = (int) Math.round(((this.CountRow(conn,
30. "susu") - 1) * ((double) (Partikel[j + 8] - 1) / (55 - 1)) + 1);
31.     }
32.     return IndeksBM;
33. }

```

Kode Program 5.4 Konversi Indeks Bahan Makanan

Berikut adalah penjelasan untuk Kode Program 5.4:

1. Baris 1 merupakan deklarasi *method* dan parameternya.
2. Baris 2-7 merupakan inisialisasi variabel-variabel yang akan digunakan dalam proses konversi indeks bahan makanan.
3. Baris 8 merupakan awal dari perulangan untuk mengubah angka permutasi dalam gen menjadi indeks bahan makanan.
4. Baris 9-30 merupakan konversi angka permutasi menjadi indeks bahan makanan sesuai dengan persamaan 4.1.

5. Baris 32 digunakan untuk mengembalikan variabel berisi indeks bahan makanan.

5.1.3.3 Konversi Nama Bahan Makanan

Proses konversi nama bahan makanan juga adalah bagian dari perhitungan *fitness* dimana angka permutasi *random* partikel yang sudah dikonversi menjadi indeks bahan makanan selanjutnya akan dikonversi lagi menjadi nama bahan makanan sesuai dengan tabel yang ada. Struktur penulisan kode program untuk proses konversi nama bahan makanan ditunjukkan pada Kode Program 5.5.

```

1. public String[] KonversiNamaBM(int[] IndeksBM) throws SQLException
2. {
3.     Koneksi koneksi = new Koneksi();
4.     Connection conn = koneksi.getKoneksi();
5.     Statement statement = conn.createStatement();
6.     int Length = IndeksBM.length;
7.     ResultSet rs;
8.     String[] NamaBM = new String[Length];
9.     for (int j = 0; j < Length; j = j + 9) {
10.         rs = this.HasilQuery1(conn, "nama_bm", "karbohidrt",
11. "indeks_bm", IndeksBM[j]);
12.         rs.next();
13.         String NamaBM1 = rs.getString("nama_bm");
14.         NamaBM[j] = NamaBM1;
15.         rs = this.HasilQuery1(conn, "nama_bm",
16. "protein_hewani", "indeks_bm", IndeksBM[j + 1]);
17.         rs.next();
18.         String NamaBM2 = rs.getString("nama_bm");
19.         NamaBM[j + 1] = NamaBM2;
20.         rs = this.HasilQuery1(conn, "nama_bm",
21. "protein_nabati", "indeks_bm", IndeksBM[j + 2]);
22.         rs.next();
23.         String NamaBM3 = rs.getString("nama_bm");
24.         NamaBM[j + 2] = NamaBM3;
25.         rs = this.HasilQuery1(conn, "nama_bm", "sayurana",
26. "indeks_bm", IndeksBM[j + 3]);
27.         rs.next();
28.         String NamaBM4 = rs.getString("nama_bm");
29.         NamaBM[j + 3] = NamaBM4;
30.         rs = this.HasilQuery1(conn, "nama_bm", "sayuranb",
31. "indeks_bm", IndeksBM[j + 4]);
32.         rs.next();
33.         String NamaBM5 = rs.getString("nama_bm");
34.         NamaBM[j + 4] = NamaBM5;
35.         rs = this.HasilQuery1(conn, "nama_bm", "buah",
36. "indeks_bm", IndeksBM[j + 5]);
37.         rs.next();
38.         String NamaBM6 = rs.getString("nama_bm");
39.         NamaBM[j + 5] = NamaBM6;
40.         rs = this.HasilQuery1(conn, "nama_bm", "lemak",
41. "indeks_bm", IndeksBM[j + 6]);
42.         rs.next();
43.         String NamaBM7 = rs.getString("nama_bm");
44.         NamaBM[j + 6] = NamaBM7;
45.         rs = this.HasilQuery1(conn, "nama_bm", "gula",
46. "indeks_bm", IndeksBM[j + 7]);
47.         rs.next();
48.         String NamaBM8 = rs.getString("nama_bm");
49.         NamaBM[j + 7] = NamaBM8;
50.         rs = this.HasilQuery1(conn, "nama_bm", "susu",
51. "indeks_bm", IndeksBM[j + 8]);
52.         rs.next();

```

53.	String NamaBM9 = rs.getString("nama_bm");
54.	NamaBM[j + 8] = NamaBM9;
55.	}
56.	return NamaBM;
	}

Kode Program 5.5 Konversi Nama Bahan Makanan

Berikut adalah penjelasan untuk Kode Program 5.5:

1. Baris 1 merupakan deklarasi *method* dan parameternya.
2. Baris 2-7 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses konversi nama bahan makanan.
3. Baris 8 merupakan permulaan perulangan untuk mengubah indeks bahan makanan menjadi nama bahan makanan.
4. Baris 9-53 merupakan konversi indeks bahan makanan menjadi nama bahan makanan sesuai dengan *database*.
5. Baris 54 berguna untuk mengembalikan variabel berisi nama bahan makanan.

5.1.3.4 Perhitungan Berat Total Bahan Makanan

Proses perhitungan berat total bahan makanan adalah bagian selanjutnya dari perhitungan *fitness* dimana dilakukan perhitungan total berat masing-masing bahan makanan. Struktur penulisan kode program untuk proses perhitungan berat total bahan makanan ditunjukkan pada Kode Program 5.6.

1.	public double[] HitungBeratTotal(double[][] Berat) {
2.	int Length = Berat.length;
3.	int JmlOrang = Berat[0].length;
4.	double[] BeratTotal = new double[Length];
5.	for (int j = 0; j < Length; j++) {
6.	for (int k = 0; k < JmlOrang; k++) {
7.	BeratTotal[j] = (BeratTotal[j] + Berat[j][k]);
8.	}
9.	}
10.	return BeratTotal;
11.	}
12.	}

Kode Program 5.6 Perhitungan Berat Total Bahan Makanan

Berikut adalah penjelasan untuk Kode Program 5.6:

1. Baris 1 merupakan deklarasi *method* dan parameternya.
2. Baris 2-4 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses perhitungan berat total bahan makanan.
3. Baris 5-6 merupakan permulaan perulangan.
4. Baris 7 merupakan penjumlahan berat masing-masing bahan makanan untuk setiap orang menjadi berat total.
5. Baris 10 digunakan untuk mengembalikan variabel berisi berat total bahan makanan.

5.1.3.5 Perhitungan Selisih Kalori

Proses perhitungan selisih kalori juga merupakan bagian dari perhitungan *fitness* dimana dilakukan perhitungan selisih jumlah kalori yang dibutuhkan dengan jumlah kalori yang dikonsumsi oleh masing-masing individu anggota keluarga. Struktur penulisan kode program untuk proses perhitungan perhitungan selisih kalori ditunjukkan pada Kode Program 5.7.

```

1. public double HitungSelisiKaloriTotal(double[] Energi, double[][]
2. Kalori) {
3.     int Popsiize = Kalori.length;
4.     int Length = Kalori.length;
5.     int JmlOrang = Kalori[0].length;
6.     //menghitung selisih kalori
7.     double[] KaloriTotal = new double[JmlOrang];
8.     double[] SelisihKalori = new double[JmlOrang];
9.     double SelisihKaloriTotal = 0;
10.    int hari = Length / 27;
11.    for (int k = 0; k < JmlOrang; k++) {
12.        for (int j = 0; j < Length; j++) {
13.            KaloriTotal[k] = KaloriTotal[k] + Kalori[j][k];
14.        }
15.        SelisihKalori[k] = Math.abs(Energi[k] - (KaloriTotal[k]
16. / hari));
17.        SelisihKaloriTotal = SelisihKaloriTotal +
18. SelisihKalori[k];
19.    }
20.    return SelisihKaloriTotal;
21. }
22.

```

Kode Program 5.7 Perhitungan Selisih Kalori

Berikut adalah penjelasan untuk Kode Program 5.7:

1. Baris 1-2 merupakan deklarasi *method* dan parameternya.
2. Baris 2-10 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses perhitungan selisih kalori.
3. Baris 11-12 merupakan permulaan perulangan.
4. Baris 13 merupakan penjumlahan kalori masing-masing bahan makanan untuk dijadikan kandungan kalori total yang akan dikonsumsi anggota keluarga.
5. Baris 15-16 digunakan untuk menghitung jumlah kalori rata-rata per hari yang akan dikonsumsi anggota keluarga.
6. Baris 17-18 digunakan untuk menghitung selisih kalori total.
7. Baris 20 digunakan untuk mengembalikan variabel yang berisi selisih kalori total.

5.1.3.6 Perhitungan Selisih Karbohidrat Total

Proses perhitungan selisih karbohidrat total juga merupakan bagian dari perhitungan *fitness* dimana dilakukan perhitungan selisih jumlah karbohidrat total yang dibutuhkan dengan jumlah karbohidrat yang dikonsumsi oleh masing-masing individu anggota keluarga. Struktur penulisan kode program untuk proses

perhitungan perhitungan selisih karbohidrat total ditunjukkan pada Kode Program 5.8.

```

1. public double HitungSelisiKarbohidratTotal(double[]
2. KebutuhanKarbohidrat, double[][] Karbohidrat) {
3.     int Length = Karbohidrat.length;
4.     int JmlOrang = Karbohidrat[0].length;
5.     double[] KarbohidratTotal = new double[JmlOrang];
6.     double[] SelisihKarbohidrat = new double[JmlOrang];
7.     double SelisihKarbohidratTotal = 0;
8.     int hari = Length / 27;
9.     for (int k = 0; k < JmlOrang; k++) {
10.        for (int j = 0; j < Length; j++) {
11.            KarbohidratTotal[k] = KarbohidratTotal[k] +
12.            Karbohidrat[j][k];
13.        }
14.        SelisihKarbohidrat[k] =
15.        Math.abs(KebutuhanKarbohidrat[k] - (KarbohidratTotal[k] / hari));
16.        SelisihKarbohidratTotal = SelisihKarbohidratTotal +
17.        SelisihKarbohidrat[k];
18.    }
19.    return SelisihKarbohidratTotal;
20. }
21.
22.
23.

```

Kode Program 5.8 Perhitungan Selisih Karbohidrat Total

Berikut adalah penjelasan untuk Kode Program 5.8:

1. Baris 1-2 merupakan deklarasi *method* dan parameternya.
2. Baris 2-8 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses perhitungan selisih karbohidrat.
3. Baris 9-10 merupakan permulaan perulangan.
4. Baris 11-12 merupakan penjumlahan karbohidrat masing-masing bahan makanan untuk dijadikan kandungan karbohidrat total yang akan dikonsumsi anggota keluarga.
5. Baris 14-15 digunakan untuk menghitung jumlah karbohidrat rata-rata per hari yang akan dikonsumsi anggota keluarga.
6. Baris 16-17 digunakan untuk menghitung selisih karbohidrat total.
7. Baris 19 digunakan untuk mengembalikan variabel yang berisi selisih karbohidrat total.

5.1.3.7 Perhitungan Selisih Protein Total

Proses perhitungan selisih protein total juga merupakan bagian dari perhitungan *fitness* dimana dilakukan perhitungan selisih jumlah protein total yang dibutuhkan dengan jumlah protein yang dikonsumsi oleh masing-masing individu anggota keluarga. Struktur penulisan kode program untuk proses perhitungan perhitungan selisih protein total ditunjukkan pada Kode Program 5.9.

```

1. public double HitungSelisiProteinTotal(double[] KebutuhanProtein,
2. double[][] Protein) {
3.     int Length = Protein.length;

```

```

4.      int JmlOrang = Protein[0].length;
5.
6.      double[] ProteinTotal = new double[JmlOrang];
7.      double[] SelisihProtein = new double[JmlOrang];
8.      double SelisihProteinTotal = 0;
9.      int hari = Length / 27;
10.     for (int k = 0; k < JmlOrang; k++) {
11.         for (int j = 0; j < Length; j++) {
12.             ProteinTotal[k] = ProteinTotal[k] + Protein[j][k];
13.         }
14.         SelisihProtein[k] = Math.abs(KebutuhanProtein[k] -
15. (ProteinTotal[k] / hari));
16.         SelisihProteinTotal = SelisihProteinTotal +
17. SelisihProtein[k];
18.     }
19.     return SelisihProteinTotal;
20. }
21.
22.

```

Kode Program 5.9 Perhitungan Selisih Protein Total

Berikut adalah penjelasan untuk Kode Program 5.9:

1. Baris 1-2 merupakan deklarasi *method* dan parameternya.
2. Baris 2-8 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses perhitungan selisih protein.
3. Baris 9-10 merupakan permulaan perulangan.
4. Baris 11-12 merupakan penjumlahan protein masing-masing bahan makanan untuk dijadikan kandungan protein total yang akan dikonsumsi anggota keluarga.
5. Baris 14-15 digunakan untuk menghitung jumlah protein rata-rata per hari yang akan dikonsumsi anggota keluarga.
6. Baris 16-17 digunakan untuk menghitung selisih protein total.
7. Baris 19 digunakan untuk mengembalikan variabel yang berisi selisih protein total.

5.1.3.8 Perhitungan Selisih Lemak Total

Proses perhitungan selisih lemak total juga merupakan bagian dari perhitungan *fitness* dimana dilakukan perhitungan selisih jumlah lemak total yang dibutuhkan dengan jumlah lemak yang dikonsumsi oleh masing-masing individu anggota keluarga. Struktur penulisan kode program untuk proses perhitungan perhitungan selisih lemak total ditunjukkan pada Kode Program 5.10.

```

1.      public double HitungSelisihLemakTotal(double[] KebutuhanLemak,
2.      double[][] Lemak) {
3.          int Length = Lemak.length;
4.          int JmlOrang = Lemak[0].length;
5.
6.          double[] LemakTotal = new double[JmlOrang];
7.          double[] SelisihLemak = new double[JmlOrang];
8.          double SelisihLemakTotal = 0;
9.          int hari = Length / 27;
10.         for (int k = 0; k < JmlOrang; k++) {
11.             for (int j = 0; j < Length; j++) {
12.                 LemakTotal[k] = LemakTotal[k] + Lemak[j][k];
13.             }

```

14.	SelisihLemak[k] = Math.abs (KebutuhanLemak[k] -
15.	(LemakTotal[k] / hari));
16.	SelisihLemakTotal = SelisihLemakTotal +
17.	SelisihLemak[k];
18.	}
19.	return SelisihLemakTotal;
20.	}
21.	

Kode Program 5.10 Perhitungan Selisih Lemak Total

Berikut adalah penjelasan untuk Kode Program 5.10:

1. Baris 1-2 merupakan deklarasi *method* dan parameternya.
2. Baris 2-8 merupakan inisialisasi variabel-variabel yang dibutuhkan dalam proses perhitungan selisih lemak.
3. Baris 9-10 merupakan permulaan perulangan.
4. Baris 11-12 merupakan penjumlahan lemak masing-masing bahan makanan untuk dijadikan kandungan lemak total yang akan dikonsumsi anggota keluarga.
5. Baris 14-15 digunakan untuk menghitung jumlah lemak rata-rata per hari yang akan dikonsumsi anggota keluarga.
6. Baris 16-17 digunakan untuk menghitung selisih lemak total.
7. Baris 19 digunakan untuk mengembalikan variabel yang berisi selisih lemak total.

5.1.3.9 Perhitungan Harga Total

Proses perhitungan harga total juga diperlukan dalam perhitungan nilai *fitness*. Untuk perhitungan harga total langkah pertama adalah menghitung harga setiap bahan makanan sesuai dengan berat keseluruhan. Struktur penulisan kode program untuk proses perhitungan perhitungan harga total ditunjukkan pada Kode Program 5.11.

1.	public double HitungHargaTotalBM(double[] Harga) {
2.	//menghitung harga total
3.	int Length = Harga.length;
4.	double HargaTotal = 0;
5.	
6.	for (int j = 0; j < Length; j++) {
7.	HargaTotal = HargaTotal + Harga[j];
8.	}
9.	return HargaTotal;
10.	}

Kode Program 5.11 Perhitungan Harga Total

5.1.3.10 Perhitungan Variasi

Perhitungan variasi juga merupakan salah satu proses dari perhitungan *fitness*. Tujuan dari penghitungan variasi adalah untuk mengetahui banyaknya variasi dari bahan makanan. Jika tidak terjadi duplikasi atau kesamaan bahan makanan dalam satu partikel maka variasi bernilai 1. Namun jika terjadi duplikasi atau kesamaan untuk bahan makanan tersebut dinilai 0. Struktur penulisan kode program untuk proses perhitungan perhitungan harga total ditunjukkan pada Kode Program 5.12.

```

1. public int HitungVariasi(String[] NamaBM) {
2.     int Length = NamaBM.length;
3.     int Variasi = 0;
4.
5.     for (int j = 0; j < Length; j++) {
6.         int Beda = 0;
7.         for (int k = 0; k < Length; k++) {
8.             if (!NamaBM[j].equalsIgnoreCase(NamaBM[k])) {
9.                 Beda = Beda + 1;
10.            }
11.        }
12.        if (Beda == (Length - 1)) {
13.            Variasi = Variasi + 1;
14.        }
15.    }
16.    return Variasi;
17. }

```

Kode Program 5.12 Perhitungan Variasi

5.1.4 Perhitungan *Pbest*

Proses perhitungan *Pbest* adalah proses untuk melihat nilai *fitness* yang terakhir dicapai adalah nilai *fitness terbesar* yang dicapai oleh partikel yang menandakan posisi terbaik dari suatu partikel pada iterasi *i*. Struktur penulisan kode program untuk proses pengecekan *pbest* ditunjukkan pada Kode Program 5.13.

```

1 public int[][] PerbandinganPbest(int indeks, int Swarmsize, int
2 Jumlahgen, double [][]Fitness, int [][][]Populasi, int [][]Pbest,
3 double[] Fitnesspbest) {
4     for (int m = 0; m < Swarmsize; m++){
5         if (Fitness[indeks][m] > Fitnesspbest[m]){
6             for (int j = 0; j < Jumlahgen; j++) {
7                 Pbest[m][j] = Populasi[indeks][m][j];
8             }
9         }
10        if (Fitness[indeks][m] < Fitness[indeks-1][m]){
11            for (int j = 0; j < Jumlahgen; j++) {
12                Pbest[m][j] = Pbest[m][j];
13            }
14        }
15        if (Fitness[indeks][m] == Fitness[indeks-1][m]){
16            for (int j = 0; j < Jumlahgen; j++) {
17                Pbest[m][j] = Populasi[indeks][m][j];
18            }
19        }
20    }
21    }
22    return Pbest;
23 }

```

Kode Program 5.13 Perhitungan *Pbest*

Berikut adalah penjelasan untuk Kode Program 5.13:

1. Baris 1-3 merupakan deklarasi *method* dan parameternya.
2. Baris 4 adalah untuk mengulang pengecekan sampai dengan sebanyak *swarm-size* yang diinputkan.

3. Baris 5-8 adalah kondisi perbandingan *fitness* partikel saat ini dengan pbest sebelumnya dimana *fitness* partikel saat ini lebih besar daripada pbest sebelumnya.
4. Baris 10-14 adalah kondisi perbandingan *fitness* partikel saat ini dengan pbest sebelumnya dimana *fitness* partikel saat ini lebih kecil daripada pbest sebelumnya.
5. Baris 15-19 adalah kondisi perbandingan *fitness* partikel saat ini dengan pbest sebelumnya dimana *fitness* partikel saat ini sama dengan pbest sebelumnya.
6. Baris 22 adalah proses mengembalikan nilai Pbest dari hasil perbandingan.

5.1.5 Perhitungan Gbest

Proses perhitungan *Gbest* adalah proses pengambilan partikel dengan nilai *fitness* terbaik/terbesar dalam suatu populasi. Struktur penulisan kode program untuk proses pengecekan *Gbest* ditunjukkan pada Kode Program 5.14.

```

1      FitnessTerbaik = pso.Perangkingan(Fitness);
2
3      PopulasiSorting = new int[swarm_size][Length];
4
5      for (int j = 0; j < swarm_size; j++) {
6          for (int k = 0; k < swarm_size; k++) {
7              if (FitnessTerbaik[j] == Fitness[k]) {
8                  Ranking[j] = (k + 1);
9                  //System.out.println(k+", "+j);
10                 PopulasiSorting[j] = Populasi[k];
11             }
12         }
13     }
14
15
16

```

Kode Program 5.14 Perhitungan Gbest

Berikut adalah penjelasan untuk Kode Program 5.14:

1. Baris 1-3 merupakan deklarasi *method* dan parameternya.
2. Baris ke 3 adalah membuat objek PopulasiSorting bertipe data int
3. Baris ke 4-10 adalah iterasi inisialisasi proses pemeringkatan partikel berdasarkan *fitness* terbaik

5.1.6 Update Kecepatan Partikel

Proses update kecepatan partikel adalah proses dimana kecepatan partikel dihitung kembali berdasarkan posisi dan kecepatan dari iterasi sebelumnya. Kecepatan partikel juga dipengaruhi oleh parameter nilai bobot inersia (w) dan juga koefisien akselerasi (C_1 & C_2). Struktur penulisan kode program untuk proses update kecepatan partikel ditunjukkan pada Kode Program 5.15

```

1      public int[][] UpdateVelocity1(int Swarmsize, int Jumlahgen, int
2      velocity, int [][]PopulasiSorting, int [][]Populasiupdate, int
3      [][]UpdatePosisi, double C1, double c2, double r1, double r2,
4      double w) {
5          int[][] Vel2= new int[Swarmsize][Jumlahgen];

```

```

6      //Random random = new Random();
7      for (int i = 0; i < Swarmsize; i++) {
8          for (int j = 0; j < Jumlahgen; j++) {
9              Vel2[i][j] = (int)
10             ((w*velocity)+(C1*r1)*(Populasiupdate[i][j]-
11             UpdatePosisi[i][j])+(c2*r2)*(PopulasiSorting[i][j]-
12             UpdatePosisi[i][j])) ;
13         }
14     }
15     }
16     return Vel2;
17
18     }

```

Kode Program 5.15 Update Kecepatan Partikel

Berikut adalah penjelasan untuk Kode Program 5.15:

1. Baris 1-4 merupakan deklarasi *method* dan parameternya.
2. Baris 5 adalah proses membuat objek Vel2 untuk menyimpan nilai update kecepatan partikel
3. Baris 6-11 adalah perulangan untuk perhitungan kecepatan partikel.

5.1.7 Update Posisi Partikel

Proses update posisi partikel adalah proses mengupdate posisi partikel setelah melalui perubahan kecepatan yang dipengaruhi parameter nilai bobot inersia (ω) dan juga koefisien akselerasi (C_1 & C_2). Struktur penulisan kode program untuk proses update posisi partikel ditunjukkan pada Kode Program 5.16.

```

1      public int[][] UpdatePosisi(int Swarmsize, int Jumlahgen, int
2      [][]Vupdate, int [][]Populasi) {
3          int[][] SwarmUpdate = new int[Swarmsize][Jumlahgen];
4          Random random = new Random();
5          for (int i = 0; i < Swarmsize; i++) {
6              for (int j = 0; j < Jumlahgen; j++) {
7                  SwarmUpdate[i][j] = Populasi[i][j]+Vupdate[i][j];
8              }
9          }
10         return SwarmUpdate;
11     }

```

Kode Program 5.16 Update Posisi Partikel

1. Baris 1-2 merupakan deklarasi *method* dan parameternya.
2. Baris 3 adalah proses membuat objek SwarmUpdate untuk menyimpan nilai *update* posisi partikel
3. Baris 4-6 adalah perulangan untuk perhitungan *update* posisi partikel.

5.2 Implementasi Antarmuka

Sub bab ini akan menjelaskan mengenai hasil implementasi antarmuka program yang berdasarkan perancangan antarmuka yang telah dibahas pada bab sebelumnya.

5.2.1 Implementasi Sistem

Saat awal membuka program yang akan muncul pertama kali adalah panel input data dan parameter dan juga. Panel data keluarga panel input data dan parameter digunakan untuk memasukkan parameter dari algoritma *particle swarm optimization* dan juga data keluarga yang akan digunakan. Terdapat tombol default untuk setting parameter dan juga data berdasarkan nilai yang sudah ditentukan agar tidak perlu memasukkan data dan parameter satu per satu. Panel data keluarga berisi tabel yang menampung data keluarga yang sudah diinput.

No	Nama	Jenis Kelamin	Usia	Berat Badan	Tinggi Badan	Aktivitas
1	Ayah	L	41	63	170	Sedang
2	Ibu	P	40	60	165	Ringan
3	Anak 1	L	20	52	155	Sedang
4	Anak 2	P	16	49	150	Sedang

Gambar 5.1 Implementasi Halaman Panel Input Parameter, Input dan Data Keluarga

Panel kebutuhan gizi berisikan tabel hasil perhitungan kebutuhan gizi masing-masing individu dari data keluarga yang telah diisikan sebelumnya.

OPTIMASI BAHAN MAKANAN DENGAN ALGORITMA PARTICLE SWARM OPTIMIZATION

Parameter Particle Swarm Optimization

Wmin: 0.4
Wmax: 0.9
C1: 2
C2: 2
IterMax: 10
Jml Partikel: 5
Jml 1 Partikel: 27
Jml Hari: 1

Info

JENIS AKTIFITAS

Aktivitas Sangat Ringan = 1
Aktivitas Ringan = 2
Aktivitas Sedang = 3
Aktivitas Berat = 4

Data Keluarga

Nama:
Usia:
Berat Badan:
Tinggi Badan:
Jenis Kelamin: ☐ Laki-Laki ☐ Perempuan
Jenis Aktivitas:

Kebutuhan Gizi Keluarga

No	Nama	Kebutuhan En...	Kebutuhan Kar...	Kebutuhan Pro...	Kebutuhan Le...
1	Ayah	2638.768	352.8306	98.9538	87.95933333...
2	Ibu	2077.0	285.5875	77.8875	69.23333333...
3	Anak 1	2492.864	342.7688	93.4824	83.09546666...
4	Anak 2	2244.34	325.4293	67.3302	74.81133333...

Gambar 5.2 Implementasi Panel Kebutuhan Gizi

Panel Hasil Optimasi berisikan tabel yang menampung hasil optimasi partikel terbaik beserta nilai fitnessnya dari setiap iterasi dan juga tabel hasil optimasi untuk menampung partikel terbaik hasil optimasi.

OPTIMASI BAHAN MAKANAN DENGAN ALGORITMA PARTICLE SWARM OPTIMIZATION

Parameter Particle Swarm Optimization

Wmin: 0.4
Wmax: 0.9
C1: 2
C2: 2
IterMax: 10
Jml Partikel: 5
Jml 1 Partikel: 27
Jml Hari: 1

Info

JENIS AKTIFITAS

Aktivitas Sangat Ringan = 1
Aktivitas Ringan = 2
Aktivitas Sedang = 3
Aktivitas Berat = 4

Data Keluarga

Nama:
Usia:
Berat Badan:
Tinggi Badan:
Jenis Kelamin: ☐ Laki-Laki ☐ Perempuan
Jenis Aktivitas:

Individu Terbaik Setiap Generasi

Iterasi	Partikel	Fitness
0	5	147.6899209540367
1	5	147.6899209540367
2	5	147.6899209540367
3	5	147.6899209540367
4	5	147.6899209540367
5	4	173.94443191785285
6	4	173.94443191785285
7	4	173.94443191785285
8	4	173.94443191785285
9	4	173.94443191785285

Hasil Optimasi

Partikel	Fitness
173.94443191785285	173.94443191785285

Gambar 5.3 Implementasi Panel Hasil Optimasi

Panel keterangan bahan makanan berisikan text area yang akan menampung keterangan bahan makanan yang dihasilkan dari partikel terbaik beserta keterangan harga, kalori, karbohidrat, protein, lemak dan lain lain.

OPTIMASI BAHAN MAKANAN DENGAN ALGORITMA PARTICLE SWARM OPTIMIZATION

Parameter Particle Swarm Optimization

Wmin:

Wmax:

C1:

C2:

IterMax:

Jml Partikel:

Jml 1 Partikel:

Jml Hari:

Info

JENIS AKTIFITAS

Aktivitas Sangat Ringan = 1

Aktivitas Ringan = 2

Aktivitas Sedang = 3

Aktivitas Berat = 4

Data Keluarga

Nama:

Usia:

Berat Badan:

Tinggi Badan:

Jenis Kelamin: ☐ Laki-Laki ☐ Perempuan

Jenis Aktifitas:

Data Keluarga | **Kebutuhan Gizi Keluarga** | **Hasil Optimasi** | **Keterangan Bahan Makanan**

Rekomendasi Bahan Makanan

Hari ke-1

Makan Pagi

Gen : 1| Index Bahan Makanan : 1| Nama Bahan Makanan : Bihun|

Takaran Orang ke-1: 100.000 gram	Kandungan Kalori : 350.000 kkal	Kandungan Karbohidrat : 100.000 gram
Takaran Orang ke-2: 62.500 gram	Kandungan Kalori : 218.750 kkal	Kandungan Karbohidrat : 62.500 gram
Takaran Orang ke-3: 100.000 gram	Kandungan Kalori : 350.000 kkal	Kandungan Karbohidrat : 100.000 gram
Takaran Orang ke-4: 50.000 gram	Kandungan Kalori : 175.000 kkal	Kandungan Karbohidrat : 50.000 gram
Berat Total : 312.500 gram		
Harga Total : Rp 10937.50		

Gen : 1| Index Bahan Makanan : 1| Nama Bahan Makanan : Babat|

Takaran Orang ke-1: 30.000 gram	Kandungan Kalori : 37.500 kkal	Kandungan Karbohidrat : 30.000 gram
Takaran Orang ke-2: 30.000 gram	Kandungan Kalori : 37.500 kkal	Kandungan Karbohidrat : 30.000 gram
Takaran Orang ke-3: 30.000 gram	Kandungan Kalori : 37.500 kkal	Kandungan Karbohidrat : 30.000 gram
Takaran Orang ke-4: 30.000 gram	Kandungan Kalori : 37.500 kkal	Kandungan Karbohidrat : 30.000 gram

Gambar 5.4 Implementasi Panel Keterangan Bahan Makanan

BAB 6 PENGUJIAN DAN ANALISIS

Pada bagian ini akan dijabarkan mengenai pelaksanaan uji coba terhadap sistem yang berdasar pada bab 4 yaitu bagian perancangan uji coba, yang meliputi uji coba terhadap beberapa aspek berikut.

6.1 Hasil dan Analisis Uji coba Berdasarkan Banyaknya Partikel

Uji coba berdasarkan banyaknya partikel bertujuan untuk mengetahui hubungan antara jumlah partikel terhadap nilai *fitness*. Banyaknya partikel yang digunakan dalam uji coba ini adalah bilangan kelipatan 5, mulai 5 hingga 50. Uji coba banyaknya partikel dilakukan sebanyak 10 kali dengan parameter lain yang ditentukan.

Untuk detail parameter yang digunakan adalah sebagai berikut:

1. Jumlah Iterasi : 10
2. Siklus Hari : 7
3. Nilai ω_{min} : 0.4
4. Nilai ω_{max} : 0.9
5. Nilai C_1 : 2
6. Nilai C_2 : 2
7. Jumlah orang : 4
8. Batas angka permutasi : 1-55

6.1.1 Hasil Pengujian

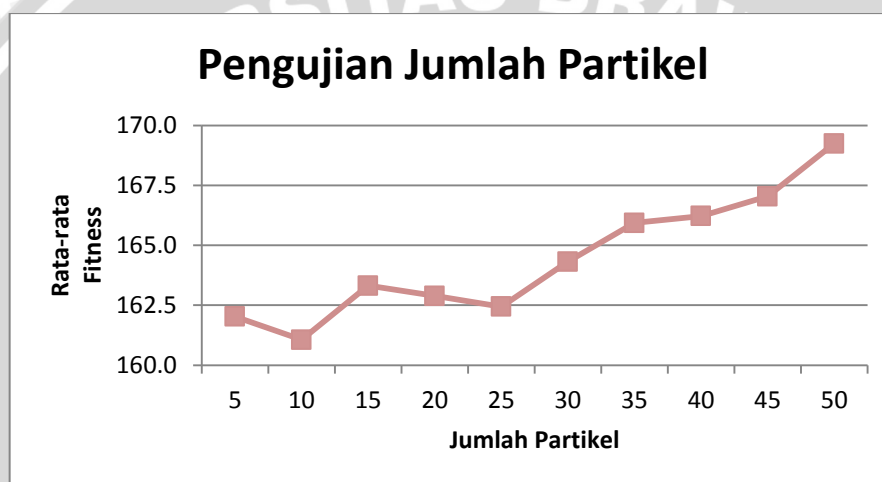
Hasil pengujian banyaknya partikel dapat dilihat pada Tabel 6.1. Terlihat dari hasil pengujian ini bahwa rata-rata nilai *fitness* terbaik didapatkan dengan jumlah partikel sebesar 50.

Tabel 6.1 Tabel Hasil Pengujian Jumlah Partikel

Jumlah Partikel	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
5	157.6	157.1	163.7	159.1	166.8	166.7	160.0	166.6	162.9	159.8	162.0
10	166.8	167.8	158.3	167.5	163.8	157.5	141.1	162.5	164.0	161.2	161.1
15	160.4	166.8	166.8	168.1	158.0	162.3	164.5	157.7	158.2	170.3	163.3
20	159.6	165.5	163.8	157.4	163.5	159.2	161.4	165.8	164.3	168.4	162.9
25	165.3	165.1	160.8	160.8	163.4	166.7	159.7	160.7	162.6	159.3	162.4
30	165.1	168.0	165.3	161.8	169.1	162.3	162.1	162.3	158.2	169.0	164.3
35	163.4	168.6	159.9	168.9	166.8	165.5	170.2	169.5	162.9	163.6	165.9
40	167.1	169.8	162.8	161.2	167.7	163.5	160.4	173.1	168.4	168.3	166.2
45	161.9	164.1	171.1	173.8	162.9	170.6	164.7	169.1	164.8	167.4	167.0
50	171.2	165.4	169.1	171.7	171.0	172.6	166.5	167.5	174.6	162.7	169.2

6.1.2 Analisis Hasil Pengujian

Pada Tabel 6.1 dan juga Gambar 6.1 menunjukkan bahwa semakin besar jumlah partikel nilai *fitness* yang dihasilkan juga semakin baik. Hal ini dikarenakan besarnya ukuran *swarm* atau banyaknya partikel memberikan variasi calon solusi yang lebih banyak dan bervariasi sehingga pencarian solusi terbaik dapat dilakukan lebih menyeluruh. Namun dengan bertambahnya jumlah partikel, waktu komputasi pun juga ikut bertambah. Waktu komputasi yang terlalu lama tentunya akan mempengaruhi kinerja sistem. Maka dari itu dari pengujian ini diambil hasil 35 partikel yang nilainya tidak berselisih terlalu jauh dengan jumlah partikel 40, 45 dan 50 agar menghasilkan *fitness* yang baik tetapi dengan waktu komputasi yang tidak terlalu lama.



Gambar 6.1 Grafik Hasil Pengujian Jumlah Partikel

6.2 Hasil dan Analisis Uji coba Berdasarkan Kombinasi Nilai ω_{min} dan ω_{max}

Pengujian kombinasi nilai bobot inersia ω_{min} dan ω_{max} digunakan untuk menentukan kombinasi nilai bobot inersia ω_{min} dan ω_{max} yang optimal yang juga akan menghasilkan solusi terbaik. Pengujian dilakukan 10 kali untuk masing-masing nilai kombinasi bobot inersia. Rentang yang digunakan adalah antara 0.4 sampai dengan 0.9 karena dianggap optimal.

Untuk detail parameter yang digunakan adalah sebagai berikut:

1. Jumlah Iterasi : 10
2. Siklus Hari : 7
3. Jumlah Partikel : 35
4. Nilai C_1 : 2
5. Nilai C_2 : 2
6. Jumlah orang : 4
7. Batas angka permutasi : 1-55

6.2.1 Hasil Pengujian

Hasil pengujian kombinasi nilai ω_{min} dan ω_{max} dapat dilihat pada Tabel 6.2. Terlihat dari hasil pengujian ini bahwa rata-rata nilai *fitness* terbaik didapatkan dengan kombinasi nilai ω_{min} dan ω_{max} sebesar 0.4 untuk nilai ω_{min} dan 0.7 untuk nilai ω_{max} .

Tabel 6.2 Tabel Hasil Pengujian Kombinasi Nilai ω_{min} dan ω_{max}

$\omega_{min};$ ω_{max}	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
0,4;0,5	158.2	161.1	164.5	164.3	159.2	164.1	163.5	161.0	162.0	165.2	162.3
0,4;0,6	160.8	160.4	164.3	159.9	164.0	163.9	161.8	163.0	161.2	164.2	162.4
0,4;0,7	163.6	168.0	159.8	167.3	160.9	164.0	162.1	161.4	162.9	167.3	163.7
0,4;0,8	164.2	157.8	164.5	165.0	159.4	170.9	160.1	156.0	162.3	163.8	162.4
0,4;0,9	162.7	160.6	164.7	160.3	162.6	163.9	166.3	165.0	161.1	163.7	163.1

6.2.2 Analisis Hasil Pengujian

Pada Tabel 6.2 dan juga Gambar 6.2 dapat dilihat bahwa nilai *fitness* terbaik didapatkan dari kombinasi nilai ω_{min} dan ω_{max} sebesar 0.4 dan 0.7. Nilai bobot inersia akan mengecil bersamaan dengan bertambahnya iterasi. ω_{min} dan ω_{max} digunakan untuk menghitung nilai ω . Semakin besar rentang selisih nilai ω_{min} dan ω_{max} maka nilai ω juga akan semakin besar. Nilai ω yang besar berdampak pada bertambahnya kecepatan partikel sehingga partikel akan bergerak pesat menuju posisi baru. Jika nilai ω kecil, maka kecepatan partikel akan menurun dan menyebabkan partikel tidak dapat mengeksplorasi swarm secara menyeluruh dan terlalu cepat menghasilkan solusi optimal. Berdasarkan pengujian nilai ω_{min} dan ω_{max} sebesar 0.4 dan 0.7 dapat dihitung sebagai berikut. Misal untuk iterasi 1, berdasarkan Persamaan 2-2, maka perhitungan nilai ω adalah sebagai berikut:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter\ max} \times iter$$

$$\omega = 0.7 - \frac{0.7 - 0.4}{10} \times 1$$

$$\omega = 0.67$$

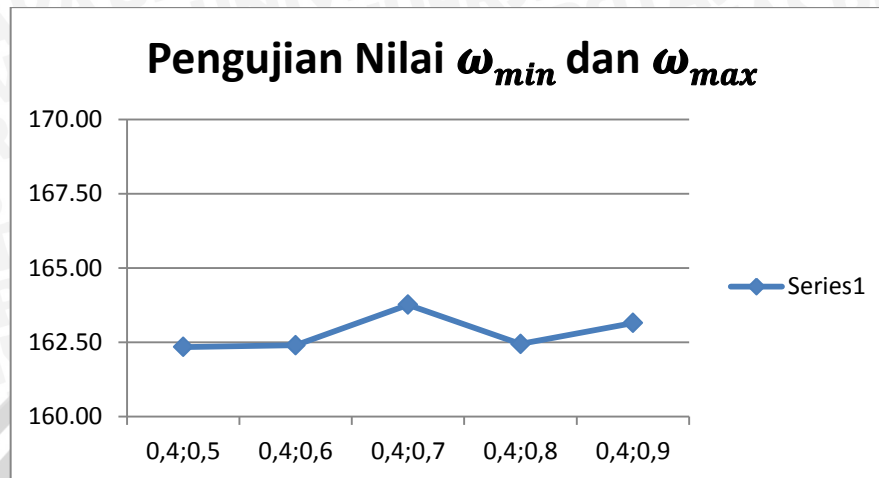
Sedangkan untuk iterasi 2, maka perhitungan nilai ω adalah sebagai berikut:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter\ max} \times iter$$

$$\omega = 0.7 - \frac{0.7 - 0.4}{10} \times 2$$

$$\omega = 0.64$$

Perhitungan ini menunjukkan rentang nilai ω yang dihasilkan dengan kombinasi nilai ω_{min} dan ω_{max} sebesar 0.4 dan 0.7 berada dalam rentang 0-1 yang bisa dikatakan baik untuk mengontrol kemampuan eksplorasi partikel.



Gambar 6.2 Grafik Hasil Pengujian Kombinasi Nilai ω_{min} dan ω_{max}

6.3 Hasil dan Analisis Uji Coba Berdasarkan Banyaknya Iterasi

Uji coba berdasarkan banyaknya iterasi bertujuan untuk mengetahui hubungan antara jumlah iterasi terhadap nilai *fitness*. Nilai yang diujikan adalah bilangan kelipatan 10 mulai 10 hingga 100.

Untuk detail parameter yang digunakan adalah sebagai berikut:

1. Jumlah Partikel : 35
2. Siklus Hari : 7
3. Nilai ω_{min} : 0.4
4. Nilai ω_{max} : 0.7
5. Nilai C_1 : 2
6. Nilai C_2 : 2
7. Jumlah orang : 4
8. Batas angka permutasi : 1-55

6.3.1 Hasil Pengujian

Hasil pengujian banyaknya iterasi dapat dilihat pada Tabel 6.3. Terlihat dari hasil pengujian ini bahwa rata-rata nilai *fitness* terbaik didapatkan dengan jumlah iterasi sebesar 100.

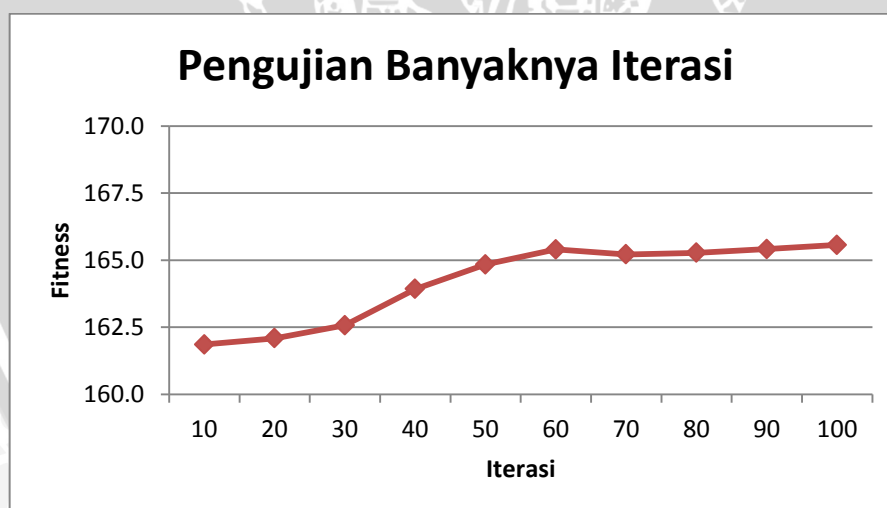
Tabel 6.3 Tabel Hasil Pengujian Banyaknya Iterasi

Iterasi	Nilai <i>Fitness</i>										Rata- Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
10	164.5	168.8	157.0	159.1	159.4	158.7	162.6	163.1	162.7	162.6	161.9
20	162.0	162.6	162.3	160.9	161.7	166.8	156.1	156.5	165.7	166.1	162.1
30	164.1	164.9	164.8	162.4	158.5	159.5	160.9	165.2	165.6	159.9	162.6

40	163.4	161.9	164.5	159.7	163.4	166.8	161.8	162.6	165.4	169.7	163.9
50	161.7	163.6	167.4	165.4	166.6	162.0	163.5	167.5	165.2	165.6	164.8
60	163.7	165.2	166.9	169.2	166.3	164.1	165.3	164.2	163.4	165.6	165.4
70	161.4	167.6	159.9	168.7	166.8	165.5	168.2	167.5	162.9	163.6	165.2
80	163.4	168.6	161.9	166.3	166.8	165.5	166.2	167.5	162.9	163.6	165.3
90	162.6	167.6	164.9	165.7	166.6	166.6	167.4	162.0	165.2	165.6	165.4
100	165.6	165.2	163.4	165.2	165.6	167.5	161.2	164.3	168.6	169.2	165.6

6.3.2 Analisis Hasil Pengujian

Pada Tabel 6.3 dan juga Gambar 6.3 menunjukkan bahwa semakin besar jumlah iterasi nilai *fitness* yang dihasilkan juga semakin baik. Hal ini dikarenakan semakin besar jumlah iterasi yang digunakan maka pencarian solusi terbaik akan terus dilakukan sampai mencapai hasil maksimum sehingga pencarian solusi terbaik dapat dilakukan lebih menyeluruh. Namun dengan dengan semakin banyaknya jumlah iterasi, perubahan posisi partikel menjadi tidak begitu signifikan pada jumlah iterasi yang terlalu besar sehingga bisa saja muncul solusi yang optimal yang hampir sama dengan jumlah iterasi berbeda. Maka dari itu dari pengujian ini diambil hasil 60 iterasi yang memiliki nilainya *fitness* yang baik dan juga dengan waktu komputasi yang tentunya lebih singkat. Jumlah partikel di atasnya tidak dipilih karena hasil *fitness* yang dihasilkan tidak bernilai terlalu jauh dibanding dengan 60 iterasi karena terjadinya konvergensi *fitness* partikel yang disebabkan tidak adanya perubahan posisi partikel yang signifikan.



Gambar 6.3 Grafik Hasil Pengujian Banyaknya Iterasi

6.4 Hasil dan Analisis Uji Coba Berdasarkan Batas Angka Permutasi

Uji coba berdasarkan batas angka permutasi bertujuan untuk mengetahui apakah batas atas angka permutasi memiliki pengaruh terhadap nilai *fitness* yang akan dihasilkan. Nilai batas angka permutasi yang diujikan adalah bilangan mulai 55 hingga 145 dengan kelipatan 10.

Untuk detail parameter yang digunakan adalah sebagai berikut:

1. Jumlah Partikel : 35
2. Siklus Hari : 7
3. Nilai ω_{min} : 0.4
4. Nilai ω_{max} : 0.7
5. Nilai C_1 : 2
6. Nilai C_2 : 2
7. Jumlah orang : 4
8. Jumlah Iterasi : 60

6.4.1 Hasil Pengujian

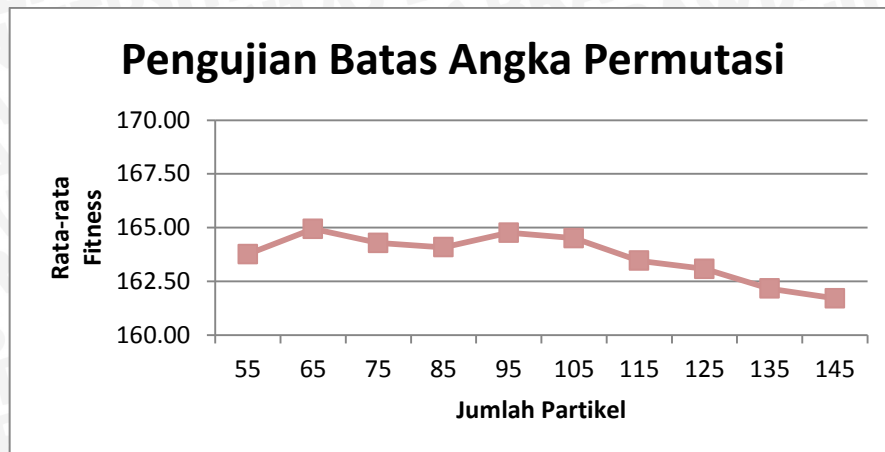
Hasil pengujian batas angka permutasi dapat dilihat pada Tabel 6.4. Terlihat dari hasil pengujian ini bahwa rata-rata nilai *fitness* terbaik didapatkan dengan batas angka permutasi sebesar 65.

Tabel 6.4 Tabel Hasil Pengujian Batas Angka Permutasi

Range	Nilai <i>Fitness</i>										Rata-Rata <i>Fitness</i>
	Pengujian ke-i										
	1	2	3	4	5	6	7	8	9	10	
55	163.6	168.0	159.8	167.3	160.9	164.0	162.1	161.4	162.9	167.3	163.7
65	163.7	163.2	166.8	173.6	161.2	163.1	165.3	163.1	163.4	165.6	164.9
75	163.7	161.0	166.0	167.0	166.6	159.5	168.0	163.4	161.7	165.5	164.2
85	167.9	159.6	162.2	160.6	169.8	162.2	167.7	167.7	159.1	163.6	164.0
95	165.5	165.2	160.4	165.1	160.5	167.5	161.2	164.2	168.5	169.1	164.7
105	169.3	169.4	165.8	158.2	163.4	174.9	160.9	158.4	161.6	162.6	164.5
115	159.5	170.4	165.6	164.9	162.5	165.3	165.4	162.4	158.1	159.8	163.4
125	161.1	158.1	157.3	161.7	176.4	162.1	165.7	163.8	159.4	164.6	163.0
135	168.3	160.8	165.2	162.8	169.2	160.5	158.2	162.2	158.7	155.2	162.1
145	165.2	160.8	165.2	162.8	163.7	160.5	158.2	162.2	162.8	155.2	161.7

6.4.2 Analisis Hasil Pengujian

Pada Tabel 6.4 dan juga Gambar 6.4 menunjukkan bahwa semakin besar batas angka permutasi nilai *fitness* yang dihasilkan semakin kecil. Hal ini dikarenakan semakin besar batas angka permutasi yang digunakan maka partikel yang dihasilkan akan banyak yang mengandung nilai lebih besar dari jumlah terbesar indeks bahan makanan dan akan berpengaruh pada saat konversi ke indeks bahan makanan dan selanjutnya mempengaruhi perhitungan jumlah variasi yang dihasilkan. Berdasarkan hasil pengujian ini, maka diambil batas angka permutasi senilai 65 yang tidak terlalu jauh dengan jumlah bahan makanan terbanyak karena memiliki sebaran variasi yang lebih banyak dan memiliki *fitness* paling baik.



Gambar 6.4 Grafik Hasil Pengujian Batas Angka Permutasi

6.5 Analisis Global Hasil Pengujian

Berdasarkan hasil pengujian yang sudah dilakukan, didapatkan parameter-parameter yang dianggap optimal sebagai berikut:

1. Jumlah Partikel : 35
2. Nilai ω_{min} : 0.4
3. Nilai ω_{max} : 0.7
4. Nilai C_1 : 2
5. Nilai C_2 : 2
6. Jumlah Iterasi : 60
7. Batas Angka Permutasi : 65

Parameter-parameter ini selanjutnya akan diujikan pada data keluarga aktual yang dihimpun melalui kuesioner.

Pada Tabel 6.5 ditampilkan data keluarga yang telah dihimpun melalui kuesioner.

Tabel 6.5 Data Keluarga Berdasarkan Kuesioner

Keluarga Ke -	Nama	Jenis Kelamin	Usia	Tinggi Badan	Berat Badan	Pekerjaan
1	Nenek	P	77	155	55	IRT
	Ibu	P	55	150	50	Guru
	Anak 1	L	29	165	63	Wiraswasta
	Anak 2	P	22	150	46	Mahasiswa
2	Ayah	L	51	173	78	PNS
	Ibu	P	50	170	72	Dosen
	Anak 1	P	21	155	52	Mahasiswa
	Anak 2	P	18	163	75	Mahasiswa

Pada Tabel 6.6 ditunjukkan data konsumsi keluarga beserta biaya dalam satu hari.

Tabel 6.6 Data Konsumsi Keluarga Beserta Biaya dalam Satu Hari.

Keluarga Ke -	Biaya Konsumsi Sehari (Rp.)	Jumlah Lauk Hewani	Jumlah Lauk Nabati	Sayuran	Buah	Susu
1	75000	2	2	3	2-3 hari sekali	memiliki lansi namun tidak mengkonsumsi susu
2	250000	2	3	3	2-3 hari sekali	memiliki lansi namun tidak mengkonsumsi susu

Pada Tabel 6.7 ditunjukkan kebutuhan gizi keluarga aktual dari Tabel 6.6.

Tabel 6.7 Kebutuhan Gizi Keluarga Aktual

Keluarga Ke -	Nama (kkal)	Energi (gr)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)
1	Nenek	1623.315	223.206	60.874	54.111
	Ibu	1867.45	256.774	70.029	62.248
	Anak1	2629.88	361.609	98.62	87.663
	Anak2	2219.86	305.231	83.245	73.995
2	Ayah	2610.59	358.956	97.897	87.02
	Ibu	2262.36	311.074	84.838	75.412
	Anak1	2227.85	306.329	83.544	74.262
	Anak2	2939.804	347.102	79.793	79.793

Berdasarkan bahan makanan yang direkomendasikan sistem, didapat selisih kandungan gizi dalam setiap bahan makanan yang direkomendasikan sistem dengan kebutuhan gizi actual keluarga seperti ditunjukkan pada Tabel 6.8.

Tabel 6.8 Kandungan Gizi Makanan Hasil Rekomendasi Sistem

Keluarga Ke -	Nama (kkal)	Energi (gr)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)
1	Nenek	1705.155	234.458	63.943	56.838
	ibu	1949.05	267.994	73.089	64.968
	anak1	2738.383	376.527	102.689	91.279
	anak2	2147.439	295.272	80.528	71.581
2	ayah	2907.168	399.735	109.018	96.905
	ibu	2247.059	308.97	84.264	74.901
	anak1	2268.65	311.939	85.074	75.621
	anak2	2692.46	390.406	80.773	89.748

Tabel 6.9 Selisih Kebutuhan Gizi Keluarga dengan Kandungan Gizi Makanan Hasil Rekomendasi Sistem

Keluarga Ke -	Nama	Selisih Energi (kkal)	Selisih Karbohidrat (gr)	Selisih Protein (gr)	Selisih Lemak (gr)
1	nenek	81.84 (5%)	11.252 (5%)	3.069 (5%)	2.727 (5%)
	ibu	81.6 (4.3%)	11.22 (4.36%)	3.06 (4.3%)	2.72 (4.3%)
	anak1	108.503 (4.1%)	14.918 (4.1%)	4.069 (4.6%)	3.616 (4.1%)
	anak2	-72.421 (-3.2%)	-9.959 (-3.2%)	-2.717 (-3.2%)	-2.414 (-3.2%)
Rata-Rata Selisih		49.880 (4.08%)	6.857 (2.5%)	1.870 (2.6%)	6.649 (2.5%)
2	ayah	296.578 (11.3%)	40.779 (11.3%)	11.121 (11%)	9.885 (11%)
	ibu	-15.301 (-0.7%)	-2.104 (-0.6%)	-0.574 (-0.6%)	-0.511 (-0.6%)
	anak1	40.8 (1.8%)	5.61 (1.8%)	1.53 (1.8%)	1.359 (1.8%)
	anak2	-247.344 (-8.4%)	43.304 (12.4%)	0.98 (1.2%)	9.955 (12.4%)
Rata-Rata Selisih		18.683 (4%)	21.897 (6.2%)	3.287 (3.35%)	5.172 (6.1%)

Pada Tabel 6.9 rata-rata selisih dihitung berdasarkan rata-rata selisih kebutuhan gizi aktual anggota keluarga. Berdasarkan Tabel 6.9, untuk keluarga ke-1 rata-rata selisih energi 4.08%, rata-rata selisih karbohidrat 2,5%, rata-rata selisih protein 2.6%, dan rata-rata selisih lemak 2.5%, dan untuk keluarga ke-2, rata-rata selisih energi 4%, rata-rata selisih karbohidrat 6.2%, rata-rata selisih protein -3.35%, dan rata-rata selisih lemak 6.1%. Menurut pakar, batas toleransi selisih kebutuhan gizi dengan kandungan gizi yang dikonsumsi adalah $\pm 10\%$. Sehingga pada keluarga ke-1 dan keluarga ke-2 kandungan energi, karbohidrat, dan protein masih memenuhi standar pakar. Adanya kandungan gizi yang tidak memenuhi standar pakar bisa saja terjadi dikarenakan luasnya ruang pencarian. Seperti yang disebutkan di atas, bahan makanan yang digunakan pada penelitian ini sebanyak 178 dan algoritma yang digunakan bersifat stokastik, sehingga proses inisialisasi solusi awal yang dilakukan kemungkinan masih jauh dari solusi terbaik.

Tabel 6.10 Biaya Konsumsi dan Variasi Makanan Sehari Hasil Rekomendasi Sistem

Keluarga ke -	biaya satu hari (Rp)	Jumlah lauk hewani	Jumlah lauk nabati	jumlah sayuran	buah	susu
1	68228.01	2	2	3	setiap hari	ya untuk usia > 49 tahun
2	75981.76	3	3	3	setiap hari	ya untuk usia > 49 tahun

Berdasarkan Tabel 6.10, maka pada keluarga ke-1, biaya konsumsi sehari dapat dihemat sebesar Rp 6771.99 atau 9.02% dengan variasi lauk hewani yang lebih banyak serta adanya buah-buahan setiap hari dan susu untuk anggota keluarga berusia di atas 49 tahun. Sedangkan untuk keluarga ke-2, biaya konsumsi sehari hasil rekomendasi sistem dapat menghemat sebesar Rp 174018.24 atau 69.6% dengan buah-buahan yang tersedia setiap hari dan juga susu untuk anggota keluarga di atas 49 tahun. Rata-rata penghematan biaya konsumsi sehari yang diperoleh dari pengujian terhadap kedua keluarga adalah sebesar 39.31%.



BAB 7 PENUTUP

7.1 Kesimpulan

Berikut ini adalah beberapa kesimpulan yang bisa didapat dari tahapan-tahapan penelitian yaitu perancangan, implementasi dan pengujian yang telah dilakukan:

1. Dalam penyelesaian permasalahan Optimasi Pemenuhan Kebutuhan Gizi Keluarga, penerapan algoritma *particle swarm optimization* dilakukan seperti alur kerja umum dari parameter tersebut. Dengan representasi partikel berisi angka *random* dari 1-55 yang nantinya di normalisasi menjadi indeks bahan makanan dan faktor yang dipertimbangkan dalam mencari nilai *fitness* adalah kebutuhan gizi yang tercukupi, harga makanan minimum dan variasi makanan yang banyak. Parameter algoritma Particle swarm optimization yang tepat adalah jumlah partikel sebanyak 35, ω_{min} sebesar 0.4, ω_{max} sebesar 0.7, C_1 sebesar 2, C_2 sebesar 2 dan jumlah iterasi sebanyak 60.
2. Dengan nilai parameter jumlah partikel= 35, $\omega_{min}= 0.4$, $\omega_{max}= 0.7$, $C_1=2$, $C_2=2$ dan jumlah iterasi= 60, hasil dari implementasi algoritma *Particle swarm optimization* untuk pemenuhan gizi keluarga adalah rekomendasi susunan bahan makanan untuk jangka waktu 7 hari dengan frekuensi 3 kali makan dalam sehari. Kualitas solusi yang dihasilkan diukur dari nilai *fitness* dari masing-masing partikel. Semakin tinggi nilai *fitness* maka semakin baik pula kualitas solusi yang dihasilkan. Dalam penyelesaian kasus aktual pada dua keluarga menghasilkan rekomendasi susunan bahan makanan yang dapat menghemat biaya konsumsi keluarga tersebut rata-rata sebesar 39.31% dengan variasi makanan yang lebih beragam. Sedangkan kandungan gizi yang dihasilkan masih dalam batas toleransi dan dapat mencukupi kebutuhan energi, karbohidrat, lemak dan protein. Dari pengujian penyelesaian kasus aktual, diketahui bahwa hasil rekomendasi dari sistem mampu memenuhi kebutuhan gizi keluarga dengan selisih yang masih berada dalam toleransi selisih kebutuhan gizi yaitu $\pm 10\%$ dan juga mampu menghemat biaya pengeluaran untuk konsumsi sebanyak 39.31%.

7.2 Saran

1. Koefisien akselerasi C_1 dan C_2 pada penelitian ini menggunakan nilai konstan sehingga memungkinkan pencarian solusi tidak mencapai performa optimal. Penelitian selanjutnya dapat menggunakan nilai koefisien akselerasi C_1 dan C_2 yang disesuaikan setiap iterasinya untuk mencapai performa pencarian solusi yang optimal.
2. Hasil rekomendasi bahan makanan dari sistem pada penelitian ini hanya berlaku untuk keluarga yang tidak menderita penyakit apapun atau memiliki riwayat keturunan penyakit. Penelitian selanjutnya dapat mengembangkan untuk orang atau keluarga dengan penyakit yang spesifik.
3. Zat gizi yang dioptimasi adalah zat gizi makro seperti karbohidrat, protein dan lemak. Untuk penelitian selanjutnya, dapat ditambahkan zat gizi yang

lebih kompleks seperti kalsium, vitamin, zat besi dan zat gizi mikro lainnya agar dapat memenuhi kebutuhan gizi dengan lebih akurat.

4. Pada penelitian ini dilakukan optimasi terhadap bahan makanan, maka dari itu penelitian selanjutnya dapat mengembangkan optimasi terhadap menu makanan agar hasil yang diberikan lebih mudah diterapkan dalam kehidupan sehari-hari dan juga mempertimbangkan kecocokan antara bahan makanan satu dengan yang lain.



DAFTAR PUSTAKA

- Kementrian Kesehatan RI. 2014.** Pedoman Gizi Seimbang. *Kementrian Kesehatan RI.* [Online] 2014. [Cited: April 20, 2016.] <http://gizi.depkes.go.id/download/Pedoman%20Gizi/PGS%20Ok.pdf>.
- Afandie, Mochamad Noor, Cholissodin, Imam and Supianto, Ahmad Afif. 2014.** *Implementasi Metode K-Nearest Neighbor untuk Pendukung Keputusan Menu Makanan Sehat dan Bergizi MASSA TUBUH (IMT) Dengan Gizi Seimbang.* Malang : Universitas Brawijaya, 2014.
- Almatsier, Sunita. 2009.** *Prinsip Dasar Ilmu Gizi.* 2009.
- . **2010.** *Prinsip Dasar ILMU GIZI.* 9th. Jakarta : PT Gramedia Pustaka Utama, 2010.
- Chen, et al H.-L. 2011.** A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest. 2011, pp. 1349-1359.
- Depkes.** Pedoman Praktis Untuk Mempertahankan Berat Badan Normal Berdasarkan Indeks Massa Tubuh. *gizi.depkes.go.id.* [Online] [Cited: September 30, 2014.]
- . **2014.** Permenkes Tentang Angka Kecukupan Gizi. [Online] 2014. [Cited: September 27, 2015.] <http://gizi.depkes.go.id/>.
- DKP, Dewan Ketahanan Pangan. 2009.** Indonesia Tahan Pangan dan Gizi 2015. *Badan Ketahanan Pangan dan Penyuluhan Pertanian Pemerintah Kabupaten Bangka.* [Online] 2009. [Cited: April 18, 2016.] bkp.bangka.go.id/donlot/tahan-pangan-dan-gizi-2015_datastudi.pdf.
- Fukuyama, Y. 2007.** *Fundamentals of Particle Swarm Optimization Techniques.* s.l. : K.Y. Lee & M.A. El-Sharkawi, John Wiley & Sons, Inc, Hoboken, NJ, USA, 2007.
- Hartono, Andry. 2006.** *Terapi Gizi dan Diet Rumah Sakit Ed. 2.* s.l. : EGC, 2006.
- Haupt, R.L., Haupt, S.E. 2004.** *Practical Genetic Algorithm, Second Edition.* s.l. : Wiley, 2004.
- Instalasi Gizi Perjan RS Dr. Cipto Mangunkusumo dan Asosiasi Dietisien Indonesia. 2010.** *Penuntun Diet edisi baru.* 25th. Jakarta : Gramedia Pustaka Utama, 2010.
- Kementrian Kesehatan. 2013.** Peraturan Menteri Kesehatan No. 75 Tahun 2013. *gizi.depkes.co.id.* [Online] 2013. [Cited: September 30, 2015.] <http://gizi.depkes.go.id/download/Kebijakan%20Gizi/PMK%2075-2013.pdf>.
- Kurniawan, E.P. & M., Z.Z. 2010.** *Fuzzy Membership Function Generation Using Particle Swarm Optimization.* s.l. : International Journal Open Problems Computation Math, No 3, 2010.
- M.Dorigo, T.Stutzle. 2004.** *Ant Colony optimization.* 2004.

Mahmudy, Wayan Firdaus. 2013. Wayan Firdaus Mahmudy. *Modul Algoritma Evolusi – Semester Ganjil 2013-2014*. [Online] April 8, 2013. [Cited: September 29, 2015.] <http://wayanfm.lecture.ub.ac.id/files/2014/04/2013-Algoritma-Evolusi-Modul.pdf>.

Maickel, Tuegeh, Soeprijanto and Purnomo, Mauridhi H. 2009. MODIFIED IMPROVED PARTICLE SWARM OPTIMIZATION FOR OPTIMAL. 2009.

Naipah, Dayawati, Retno Novi and Wirayuda, Tjokorda Agung Budi. 2008. Peramalan Deret Waktu Penjualan Menggunakan Kombinasi Algoritma Genetika dan Simulated Annealing. [Online] 2008. [Cited: September 29, 2015.] <https://repository.telkomuniversity.ac.id/pustaka/files/94569/resume/peramalan-deret-waktu-penjualan-menggunakan-kombinasi-algoritma-genetika-dan-simulated-annealing.pdf>.

Nouaouria, N., Boukadoum, M. & Proulx, R. 2013. *Particle Swarm Classification : A Survey and Positioning*. s.l. : Pattern Recognition, Vol 46, 2013.

Omizegba, E.E & Adebayo, G.E. 2009. *Optimizing Fuzzy Membership Using Particle Swarm Algorithm*. San Antonio : Proceeding of IEEE International Conference in Systems, Man, and Cybernetics, IEEE, 2009.

Particle Swarm Optimization : A Tutorial. **Blondin, J. 2009.** 2009, pp. 1-5.

Permenkes. 2014. *Peraturan Menteri Kesehatan Republik Indonesia Nomor 41 Tahun 2014 Tentang Pedoman Gizi Seimbang*. 2014.

Pratiwi, Monica Intan, Mahmudy, Wayan Firdaus and Dewi, Candra. 2014. Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Gizi. 2014, pp. 1-8.

Rianawati, A. and Mahmudy, Wayan F. 2015. *Implementasi Algoritma Genetika Untuk Optimasi Komposisi Makanan Bagi Penderita Diabetes Mellitus*. s.l. : DORO, 2015.

Rismayanti and Harihayati, Tati. 2015. Implementasi Algoritma Genetika untuk Penjadwalan Mata Pelajaran di SMAN 1 Ciwidey. 2015.

Sedighizadeh, Davoud and Masehian, Ellips . 2009. Particle Swarm Optimization Methods, Taxonomy and Applications. 2009.

Supariasa, I Dewa Nyoman. 2001. *Penilaian Status Gizi*. 2001.

LAMPIRAN A TIPE-TIPE ANJURAN PORSI SEHARI

Tipe	Usia		KH		PH		PN		Buah		Sayur		Lemak		Susu		Gula	
	Minimal	Maksimal	L	P	L	P	L	P	L	P	L	P	L	P	L	P	L	P
1	1	3	3	3	1	1	1	1	1.5	1.5	3	3	3	3	1	1	2	2
2	4	6	4	4	2	2	2	2	2	2	3	3	4	4	1	1	2	2
3	7	9	4.5	4.5	2	2	3	3	3	3	3	3	5	5	1	1	2	2
4	10	12	5	4	2.5	2	3	3	3	3	4	4	5	5	1	1	2	2
5	13	15	6.5	4.5	3	3	3	3	3	3	4	4	6	5	1	1	2	2
6	16	18	8	5	3	3	3	3	3	3	4	4	6	5	0	0	2	2
7	19	29	8	5	3	3	3	3	3	3	5	5	7	5	0	0	2	2
8	30	49	7.5	4.5	3	3	3	3	3	3	5	5	6	6	0	0	2	2
9	50	64	6.5	4.5	3	3	3	3	4	4	5	5	6	4	1	1	1	2
10	65	~	5	3.5	3	3	3	3	4	4	4	4	4	4	1	1	2	2

Keterangan:

L : Laki- laki

P : Perempuan

KH : Karbohidrat

PH : Protein hewani

PN :Protein Nabati

LAMPIRAN B DAFTAR BAHAN MAKANAN

A.1 Kelompok Sumber Karbohidrat

Kode	Bahan Makanan	Harga per kg	Urt	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Bihun	35000	1/2 gelas	50	175	40	4	0
2	Bubur beras	5050	2 gls	400	175	40	4	0
3	Jagung segar	5000	3 bj sdg	125	175	40	4	0
4	Kentang	12000	2 bh sdg	210	175	40	4	0
5	Krekers	45000	5 bh besar	50	175	40	4	0
6	Makaroni	30000	1/2 gelas	50	175	40	4	0
7	Mi Basah	10000	2 gls	200	175	40	4	0
8	Mi Kering	20000	1 gls	50	175	40	4	0
9	Nasi Beras Giling	10100	3/4 gls	100	175	40	4	0
10	Nasi Ketan Hitam	16000	3/4 gls	100	175	40	4	0
11	Nasi Ketan Putih	14000	3/4 gls	100	175	40	4	0
12	Roti Putih	57500	2 iris	70	175	40	4	0
13	Roti Gandum	43715	2 iris	70	175	40	4	0
14	Singkong	2500	1 1/2 ptg	120	175	40	4	0
15	Sukun	10000	3 ptg sdg	150	175	40	4	0
16	Tape Beras Ketan	47200	5 sdm	100	175	40	4	0
17	Tape Singkong	20000	1 ptg sdg	100	175	40	4	0

A.2 Kelompok Sumber Protein Hewani

Kode	Bahan Makanan	Harga per kg	URT	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Babat	69000	1 ptg sdg	40	50	0	7	2
2	Cumi-cumi	40000	1 ekor kcl	45	50	0	7	2
3	Daging asap	200000	1 lembar	20	50	0	7	2
4	Daging Ayam tanpa kulit	24000	1 ptg sdg	40	50	0	7	2
5	Dendeng daging sapi	390000	1 ptg sdg	15	50	0	7	2
6	Gabus Kering	200000	1 ptg kcl	10	50	0	7	2
7	Ikan asin kering	28000	1 ptg sdg	15	50	0	7	2
8	Ikan kakap	80000	1/3 ekor bsr	35	50	0	7	2
9	Ikan kembung	47000	1/3 ekor sdg	30	50	0	7	2
10	Ikan lele	20000	1/2 ekor sdg	40	50	0	7	2
11	Ikan mas	49000	1/3 ekor sdg	45	50	0	7	2
12	Ikan mujair	30000	1/3 ekor kcl	30	50	0	7	2
13	Ikan peda	89000	1 ekor kcl	35	50	0	7	2
14	Ikan pindang	9000	1/2 ekor sdg	35	50	0	7	2
15	Ikan segar	26000	1 ptg sdg	40	50	0	7	2
16	Kepiting	40000	1/3 gls	50	50	0	7	2
17	Kerang	32000	1/2 gls	90	50	0	7	2
18	Putih telur ayam	18000	2 1/2 btr	65	50	0	7	2
19	Rebon kering	89000	2 sdm	10	50	0	7	2
20	Sepat kering	110000	1 ptg sdg	20	50	0	7	2
21	Teri kering	58000	1 sdm	15	50	0	7	2
22	Teri nasi	32000	1/3 gls	20	50	0	7	2
23	Udang segar	56000	5 ekor sdg	35	50	0	7	2

24	Bakso	63000	10 bj sdg	170	75	0	7	5
25	Daging kambing	108000	1 ptg sdg	40	75	0	7	5
26	Daging sapi	105000	1 ptg sdg	35	75	0	7	5
27	Hati ayam	22000	1 bh sdg	30	75	0	7	5
28	Hati sapi	55000	1 ptg sdg	35	75	0	7	5
29	Otak	30000	1 ptg bsr	35	75	0	7	5
30	Telur ayam	18000	1 btr	55	75	0	7	5
31	Telur bebek asin	1750	1 btr	50	75	0	7	5
32	Telur puyuh	27000	5 btr	55	75	0	7	5
33	Usus sapi	30000	1 ptg bsr	50	75	0	7	5
34	Bebek	45000	1 ptg sdg	45	150	0	7	13
35	Belut	40000	3 ekor kcl	50	150	0	7	13
36	Kornet Daging Sapi	76500	3 sdm	45	150	0	7	13
37	Kuning telur ayam	18000	4 btr	160	150	0	7	13
38	Sardencis	42300	1/2 ptg sdg	35	150	0	7	13
39	Sosis	84000	1/2 ptg	50	150	0	7	13
40	Telur ikan	79000	1 ptg sdg	40	150	0	7	13

A.3 Kelompok Sumber Protein Nabati

Indeks	Bahan Makanan	Harga per kg	URT	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Kacang Hijau	19000	2 sdm	20	75	7	5	3
2	Kacang Kedele	10000	2 1/2 sdm	25	75	7	5	3
3	Kacang Merah	24000	2 sdm	20	75	7	5	3
4	Kacang Mente	145000	2 1/2 sdm	15	75	7	5	3

5	Kacang Tanah	22000	2 sdm	15	75	7	5	3
6	Kacang Tanah Kupas	22000	2 sdm	15	75	7	5	3
7	Kacang Tolo	15000	2 sdm	20	75	7	5	3
8	Kembang Tahu	165000	1 lembar	20	75	7	5	3
9	Oncom	35000	2 ptg kecil	40	75	7	5	3
10	Pete Segar	10000	1/2 gls	55	75	7	5	3
11	Selai Kacang Tanah	128000	1 sdm	15	75	7	5	3
12	Tahu	10000	1 bj besar	110	75	7	5	3
13	Tempe	10000	2 ptg sdg	50	75	7	5	3

A.4 Kelompok Sayuran Jenis A

Kode	Bahan Makanan	Harga per kg	Berat (g)
1	Gambas	3000	100
2	Jamur Kuping Segar	67500	100
3	Kangkung	26666	100
4	Ketimun	6000	100
5	Labu Air		100
6	Lettuce	42900	100
7	Lobak	23900	100
8	Slada	12000	100
9	Slada air	16666	100
10	Tomat	12000	100

A.5 Kelompok Sayuran Jenis B

Kode	Bahan Makanan	Harga per kg	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Bawang bombay	20000	100	25	5	1	0
2	Bayam	12000	100	25	5	1	0
3	Bit	80000	100	25	5	1	0
4	Brokoli	17000	100	25	5	1	0
5	Buncis	7000	100	25	5	1	0
6	Cabe Merah Besar	38000	100	25	5	1	0
7	Cabe Hijau Besar	18000	100	25	5	1	0
8	Caisim	10000	100	25	5	1	0
9	Daun Bawang	13000	100	25	5	1	0
10	Daun Bluntas		100	25	5	1	0
11	Daun Kacang Panjang		100	25	5	1	0
12	Daun Kecipir		100	25	5	1	0
13	Daun Kemangi	1000	100	25	5	1	0
14	Daun Koro		100	25	5	1	0
15	Daun Lobak		100	25	5	1	0
16	Daun Lompong Tales		100	25	5	1	0
17	Daun Pakis		100	25	5	1	0
18	Genjer		100	25	5	1	0
19	Jagung Muda		100	25	5	1	0
20	Jantung Pisang		100	25	5	1	0

21	Kangkung		100	25	5	1	0
22	Kacang Buncis		100	25	5	1	0
23	Kacang Panjang	7000	100	25	5	1	0
24	Kapri Muda		100	25	5	1	0
25	Kecipir		100	25	5	1	0
26	Kembang Kol	32900	100	25	5	1	0
27	Kucai	29000	100	25	5	1	0
28	Kol	12500	100	25	5	1	0
29	Labu Siam	13500	100	25	5	1	0
30	Labu Waluh	14500	100	25	5	1	0
31	Pare	19900	100	25	5	1	0
32	Pecay		100	25	5	1	0
33	Pepaya Muda		100	25	5	1	0
34	Rebung	35600	100	25	5	1	0
35	Sawi	8300	100	25	5	1	0
36	Seledri	8500	100	25	5	1	0
37	Taoge Kacang Hijau	28750	100	25	5	1	0
38	Terong	15900	100	25	5	1	0
39	Wortel	7000	100	25	5	1	0
40	Bayam Merah	30000	100	50	10	3	0
41	Daun Katuk		100	50	10	3	0
42	Daun Ketela Rambat		100	50	10	3	0
43	Daun Labu Siam		100	50	10	3	0
44	Daun Mangkokan		100	50	10	3	0
45	Daun Melinjo		100	50	10	3	0

46	Daun Pepaya		100	50	10	3	0
47	Daun Singkong	18750	100	50	10	3	0
48	Daun Talas		100	50	10	3	0
49	Kacang Kapri		100	50	10	3	0
50	Mlinjo		100	50	10	3	0
51	Nangka Muda		100	50	10	3	0
52	Taoge Kacang Kedele	45000	100	50	10	3	0

A.6 Bahan Makanan Gula dan Buah

Kode	Bahan Makanan	Harga per kg	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Anggur	40000	125	50	12	0	0
2	Apel	26000	75	50	12	0	0
3	Arbei		135	50	12	0	0
4	Belimbing	19000	140	50	12	0	0
5	Blewah		70	50	12	0	0
6	Duku		80	50	12	0	0
7	Durian	50000	35	50	12	0	0
8	Jambu Air		110	50	12	0	0
9	Jambu Biji	17500	100	50	12	0	0
10	Jeruk Bali		105	50	12	0	0
11	Jeruk Manis	30000	110	50	12	0	0
12	Jeruk Nipis	35000	135	50	12	0	0
13	Kedondong	13000	120	50	12	0	0
14	Kesemek		65	50	12	0	0

15	Kolang-kaling		25	50	12	0	0
16	Kurma		15	50	12	0	0
17	Kiwi	48800	110	50	12	0	0
18	Leci		75	50	12	0	0
19	Mangga		90	50	12	0	0
20	Manggis		80	50	12	0	0
21	Markisa		35	50	12	0	0
22	Melon	15000	190	50	12	0	0
23	Nangka		45	50	12	0	0
24	Nanas	17000	95	50	12	0	0
25	Peach		115	50	12	0	0
26	Pear	19500	85	50	12	0	0
27	Pepaya	10000	110	50	12	0	0
28	Pisang Ambon		50	50	12	0	0
29	Pisang Mas		40	50	12	0	0
30	Pisang Raja		40	50	12	0	0
31	Plum	99000	140	50	12	0	0
32	Rambutan	23000	75	50	12	0	0
33	Salak	7500	65	50	12	0	0
34	Sawo		55	50	12	0	0
35	Semangka	6000	180	50	12	0	0
36	Sirsak	40000	60	50	12	0	0
37	Srikaya		50	50	12	0	0
38	Strawberry	20000	215	50	12	0	0
39	Gula	11800	13	50	12	0	0

40	Madu	120000	15	50	12	0	0
----	------	--------	----	----	----	---	---

A.7 Bahan Makanan Gula

Kode	Bahan Makanan	Harga per kg	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Gula pasir	11800	8	30	7.5	0	0
2	Gula palm/aren	34000	10	30	7.5	0	0
3	Madu	120000	5	30	7.5	0	0
4	Selai	37000	10	30	7.5	0	0
5	Permen	39000	20	30	7.5	0	0
6	Sirup	28000	5	30	7.5	0	0

A.8 Bahan Makanan Lemak

Kode	Bahan Makanan	Harga per kg	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Alpukat	13000	60	50	0	0	5
2	Kacang Almond	300000	10	50	0	0	5
3	Mayonnaise	30000	20	50	0	0	5
4	Minyak Jagung	103000	5	50	0	0	5
5	Minyak Kacang Kedele	89000	5	50	0	0	5
6	Minyak Zaitun	150000	5	50	0	0	5
7	Mentega	50000	5	50	0	0	5
8	Santan	33000	40	50	0	0	5

9	Kelapa	8000	15	50	0	0	5
10	Keju Krim	218000	15	50	0	0	5
11	Minyak Kelapa	26800	5	50	0	0	5
12	Minyak Kelapa Sawit	11000	5	50	0	0	5

A.9 Bahan Makanan Susu

Kode	Bahan Makanan	Harga per kg	URT	Berat (g)	Kalori	Karbohidrat	Protein	Lemak
1	Susu skim bubuk	113000	4 sdm	20	75	10	7	0
2	Yoghurt non fat	90000	2/3 gls	120	75	10	7	0
3	Keju	115000	1 ptg kcl	35	125	10	7	6
4	Susu sapi	6000	1 gls	200	125	10	7	6
5	Susu kental tak manis	23376	1/2 gls	100	125	10	7	6
6	Yoghurt susu	32000	1 gls	200	125	10	7	6
7	Susu bubuk	87500	6 sdm	30	150	10	7	10