

IMPLEMENTASI ALGORITMA GENETIKA PADA OPTIMASI BIAYA PEMENUHAN KEBUTUHAN GIZI

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

MONICA INTAN PRATIWI

105090607111011

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

PROGRAM STUDI TEKNIK INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA GENETIKA PADA OPTIMASI BIAYA PEMEMUHAN KEBUTUHAN GIZI

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

MONICA INTAN PRATIWI

105090607111011

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Wayan Firdaus Mahmudy, S.Si., MT., Ph.D
NIP. 197209191997021001

Candra Dewi, S.Kom., MSc
NIP. 197711142003122001

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA GENETIKA PADA OPTIMASI BIAYA PEMENUHAN KEBUTUHAN GIZI

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

MONICA INTAN PRATIWI

105090607111011

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 10 Juli 2014
dan dinyatakan memenuhi syarat untuk memperoleh
gelar sarjana dalam bidang Ilmu Komputer

Penguji I,

Penguji II

Nurul Hidayat, S.Pd., MSc
NIP. 196804302002121001

M. Ali Fauzi S.Kom., M.Kom

Penguji III

Satrio Agung Wicaksono, S.Kom., M.Kom
NIK. 86052106110114

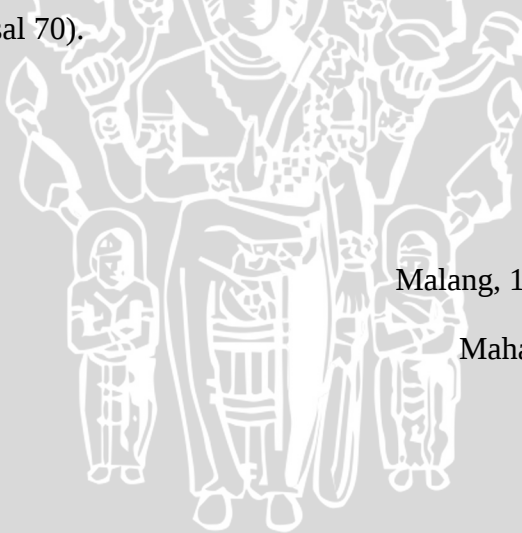
Mengetahui,
Ketua Program Studi Teknik Informatika/Ilmu Komputer

Drs. Marji, MT.
NIP. 196708011992031001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan 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, 12 Juni 2014

Mahasiswa,

Monica Intan Pratiwi
NIM. 105090607111011

KATA PENGANTAR

Syukur dan alhamdulillah penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul: “Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi”.

Skripsi ini diajukan sebagai syarat ujian skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Teknologi Informasi Dan Ilmu Komputer (PTIIK), Program Studi Informatika/Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesaikannya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Wayan Firdaus Mahmudy, S.Si., MT., Ph.D selaku Dosen Pembimbing Skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Candra Dewi S.Kom., MSc selaku Dosen Pembimbing kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Ir. Sutrisno, MT., selaku Ketua Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, MT. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Bapak Soedjoko dan Ibu Sumarmi selaku orangtua yang selalu memberikan semangat dan doa hingga terselesaikannya skripsi ini.
6. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
7. Indah Kurniawati, Meitasari Winardi, Dwy Saputro, Ayu Puspo Sari, Fayruz Al-Baity, Anjar Dwi O., Farah Bahtera P, dan semua teman-teman Teknik Informatika angkatan 2010. Terima kasih atas segala bantuannya selama menempuh studi.
8. Adam Anas Makruf yang menjadi tempat curhat saat senang dan sedih.

9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 12 Juni 2014

Penulis



ABSTRAK

Monica Intan Pratiwi. 2014 : Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi.

Dosen Pembimbing : Wayan Fidaus Mahmudy, S.Si., MT., Ph.D dan Candra Dewi, S.Kom., MSc

Masalah pemenuhan kebutuhan gizi pada manusia adalah suatu permasalahan yang sangat penting. Terdapat tiga macam unsur gizi yang dapat memberikan kalori bagi tubuh manusia, yaitu karbohidrat, lemak, dan protein. Selain itu, pengeluaran biaya juga menjadi masalah dalam pemenuhan kebutuhan gizi. Algoritma genetika adalah salah satu metode optimasi yang dapat menyelesaikan permasalahan kombinasi dalam menentukan bahan makanan yang memenuhi gizi dengan biaya minimum. Pada penelitian ini digunakan 125 data bahan makanan beserta harga dan kandungan gizinya yang akan digunakan untuk pengujian. Dari data tersebut akan dibentuk sebuah populasi dengan jumlah yang bervariasi. Setiap kromosom memiliki panjang (jumlah gen) yang berbeda, yaitu interval 5-15 gen. Masing-masing gen menyatakan indeks nomor bahan makanan. Banyak populasi yang optimal dari hasil uji coba adalah 40 populasi dengan rata-rata fitness 38950. Sedangkan banyak generasi yang optimal adalah 500 generasi dengan rata-rata fitness 38820. Nilai probabilitas *crossover* dan probabilitas mutasi didapatkan dari nilai rata-rata fitness terbaik yaitu 39030 dengan probabilitas *crossover* 0,4 dan probabilitas mutasi 0,6. Hasil akhir adalah kombinasi bahan makanan terbaik yang kandungan gizinya mencukupi dan memiliki biaya minimum.

Kata Kunci : Algoritma genetika, optimasi biaya, kebutuhan gizi

ABSTRACT

Monica Intan Pratiwi, 2014 : Genetic Algorithm Implementation at Nutrient Needs Compliance Cost Optimization.

Advisor : Wayan Fidaus Mahmudy, S.Si., MT., Ph.D and Candra Dewi, S.Kom., MSc

The compliance of nutrient needs is an important problem for humans. There are three kinds of nutrient elements that can give calorie to the human body, that are carbohydrate, fat, and protein. Moreover, cost expenses are also becoming a problem in fulfilling nutritional needs. Genetic algorithm is one of methods that can solve optimization problems in determining the food combinations that fulfill nutrition with minimum cost. This study used 125 food data with its price and nutritional content which will be used for testing. From these data will be established a population with varying amounts. Each chromosome has a different length (number of genes), that is 5-15 gene intervals. Each gene expressed index number groceries. The number of optimal population of the test results are 40 populations with 38 950 average fitness. Meanwhile the number of optimal generation is 500 generations with 38820 average fitness. The crossover rate and mutation rate values are obtained from the best of average fitness value that is 39030 with crossover probability of 0.4 and mutation probability of 0.6. The final result is a the best food combination that fullfill the nutritional content and have a minimum cost.

Keywords : Genetic algorithm, cost optimization, nutrient needs

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS SKRIPSI	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
DAFTAR <i>SOURCE CODE</i>	xiii
DAFTAR PERSAMAAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Algoritma Genetika	5
2.2 Struktur Umum Algoritma	6
2.3 Parameter-parameter Algoritma Genetika	6
2.3.1 Ukuran Populasi	7
2.3.2 Banyaknya Generasi	7
2.3.3 Probabilitas Crossover	7
2.3.4 Probabilitas Mutasi	8
2.4 Penerapan Algoritma Genetika	8
2.4.1 Membangun Generasi Awal	8
2.4.2 Representasi Kromosom	8
2.4.3 Operator Genetika	9
2.4.4 Seleksi	11

2.5 Gizi	12
2.5.1 Energi.....	12
2.6 Persamaan Kebutuhan Kalori	13
2.7 Karbohidrat.....	14
2.8 Protein.....	15
2.9 Lemak.....	16
BAB III METODOLOGI PENELITIAN DAN PERANCANGAN.....	18
3.1 Tahapan Penelitian	18
3.2 Analisa Kebutuhan Sistem	19
3.2.1 Deskripsi Umum Sistem	19
3.2.2 Data yang Digunakan.....	19
3.3 Perancangan Sistem.....	19
3.4 Proses Algoritma Genetika.....	20
3.4.1 Membuat Populasi Awal.....	22
3.4.2 Crossover	22
3.4.3 Mutasi	23
3.4.4 Menghitung Nilai Fitness.....	25
3.4.5 Seleksi.....	27
3.4.6 Perhitungan Manual.....	28
3.4.7 Inisialisasi Parameter Awal	29
3.4.8 Membuat Populasi Awal.....	30
3.4.9 Menghitung Nilai <i>Fitness</i>	32
3.4.10 Reproduksi	34
3.4.11 Evaluasi.....	36
3.4.12 Seleksi.....	38
3.4.13 Memilih Kromosom Terbaik	39
3.5 Perancangan User Interface.....	39
3.5.1 Tampilan Halaman Utama	39
3.5.2 Tampilan Halaman Hasil Proses Algoritma Genetika.....	40
3.6 Perancangan Uji Coba dan Evaluasi.....	41
3.6.1 Uji Coba Ukuran Populasi	42

3.6.2 Uji Coba Banyaknya Generasi.....	42
3.6.3 Uji Coba Kombinasi Probabilitas <i>Crossover</i> dan Mutasi.....	43
BAB IV IMPLEMENTASI	45
4.1 Lingkungan Implementasi.....	45
4.1.1 Lingkungan Perangkat Keras.....	45
4.1.2 Lingkungan Perangkat Lunak.....	45
4.2 Implementasi Program.....	45
4.2.1 Struktur Data.....	46
4.2.2 Membangkitkan Populasi Awal.....	46
4.2.3 Perhitungan <i>Fitness</i>	46
4.2.4 Perhitungan Penalti.....	47
4.2.5 Proses <i>Crossover</i>	47
4.2.6 Proses Mutasi.....	48
4.2.7 Proses Seleksi <i>Elitis</i>	49
4.2.8 Pemilihan Kromosom Terbaik.....	49
4.3 Implementasi User Interface.....	49
BAB V PENGUJIAN DAN ANALISA	51
5.1 Pengujian dan Analisa Ukuran Populasi	51
5.2 Pengujian dan Analisa Banyaknya Generasi.....	52
5.3 Pengujian dan Analisa Kombinasi Probabilitas <i>Crossover</i> dan Probabilitas Mutasi.....	53
BAB VI PENUTUP	56
6.1 Kesimpulan.....	56
6.2 Saran.....	56
DAFTAR PUSTAKA	57
LAMPIRAN DATA HARGA DAN BAHAN MAKANAN.....	59

DAFTAR GAMBAR

Gambar 2.1 Pseudo code siklus algoritma genetika.....	6
Gambar 2.2 Contoh proses crossover.....	9
Gambar 2.2 Contoh proses mutasi	10
Gambar 3.1 Diagram blok tahap penelitian	18
Gambar 3.2 Proses algoritma gnetika	20
Gambar 3.3 Proses crossover	22
Gambar 3.4 Detail proses crossover.....	23
Gambar 3.5 Proses mutasi.....	24
Gambar 3.6 Detail proses mutasi	25
Gambar 3.7 Proses perhitungan fitness.....	26
Gambar 3.8 Proses penalti.....	27
Gambar 3.9 Proses seleksi elitis.....	28
Gambar 3.10 Proses crossover	34
Gambar 3.11 Proses mutasi.....	35
Gambar 3.12 Rancangan user interface halaman utama	39
Gambar 3.13 Rancangan user interface hasil proses algoritma genetika.....	41
Gambar 4.1 Implementasi user interface halaman utama	50
Gambar 4.2 Implementasi user interface hasil proses algoritma genetika.....	50
Gambar 5.1 Grafik hasil uji coba ukuran populasi.....	51
Gambar 5.2 Grafik hasil uji coba banyaknya generasi.....	53
Gambar 5.3 Grafik hasil uji coba kombinasi probabilitas <i>crossover</i> dan mutasi...54	

DAFTAR TABEL

Tabel 2.1 Contoh representasi kromosom.....	8
Tabel 3.1 Kebutuhan gizi	29
Tabel 3.2 Total harga maksimal.....	29
Tabel 3.3 Populasi awal	31
Tabel 3.4 Contoh perhitungan kromosom ke-1	32
Tabel 3.5 Hasil perhitungan nilai fitness.....	33
Tabel 3.6 Proses <i>random addRemove</i> panjang kromosom	36
Tabel 3.7 Proses <i>addRemove</i> pada kromosom.....	36
Tabel 3.8 Hasil perhitungan <i>fitness parent</i> dan <i>offSpring</i>	37
Tabel 3.9 Hasil seleksi elitis.....	38
Tabel 3.10 Hasil kromosom terbaik.....	39
Tabel 3.11 Rancangan uji coba ukuran populasi	42
Tabel 3.12 Rancangan uji coba banyaknya generasi	43
Tabel 3.13 Rancangan uji coba kombinasi probabilitas <i>crossover</i> dan mutasi.....	44
Tabel 5.1 Hasil uji coba ukuran populasi.....	51
Tabel 5.2 Hasil uji coba banyaknya generasi.....	52
Tabel 5.3 Hasil uji coba kombinasi probabilitas <i>crossover</i> dan mutasi.....	54

DAFTAR SOURCE CODE

Source code 4.1 Struktur data	46
Source code 4.2 Membangkitkan populasi awal.....	46
Source code 4.3 Perhitungan <i>fitness</i>	47
Source code 4.4 Perhitungan penalti.....	47
Source code 4.5 Proses <i>crossover</i>	48
Source code 4.6 Proses mutasi.....	49
Source code 4.7 Proses seleksi elitis.....	49
Source code 4.8 Pemilihan kromosom terbaik.....	49



DAFTAR PERSAMAAN

Persamaan 2.1 Fitness.....	10
Persamaan 2.2 Penalti.....	10
Persamaan 2.3 Penalti _{1...4}	10
Persamaan 2.4a Angka metabolisme basal untuk laki-laki.....	12
Persamaan 2.4b Angka metabolisme basal untuk perempuan.....	12
Persamaan 2.5 Total Energy Expenditure.....	13
Persamaan 2.6 Karbohidrat.....	14
Persamaan 2.7 Protein.....	15
Persamaan 2.8 Lemak.....	17



BAB I

PENDAHULUAN

1.1 Latar Belakang

Semua manusia pasti membutuhkan gizi untuk mempertahankan hidup, menunjang pertumbuhan, dan melakukan aktifitas fisik. Menurut [MOE-98] terdapat tiga macam unsur gizi yang dapat memberikan kalori bagi tubuh manusia, yaitu karbohidrat, lemak, dan protein. Agar bisa mendapatkan gizi yang optimal, tubuh manusia perlu mengkonsumsi makanan sehari-hari yang mengandung zat gizi seimbang. Namun terkadang terdapat permasalahan dalam pengeluaran biaya. Pada saat biaya kebutuhan pokok naik, harga bahan makanan pun akan ikut naik. Hal tersebut akan mempengaruhi pengeluaran biaya dalam memenuhi kebutuhan gizi.

Pedoman Umum Gizi Seimbang (PUGS) adalah pedoman dasar tentang gizi seimbang yang disusun sebagai penuntun pada perilaku konsumsi makanan di masyarakat secara baik dan benar. Bahan makanan dikelompokkan berdasarkan tiga fungsi utama zat besi, yaitu sumber energi atau tenaga, sumber protein, dan sumber zat pengatur berupa sayuran buah. PUGS menganjurkan agar 60-75% kebutuhan energi diperoleh dari karbohidrat, 10-15% dari protein dan 10-15% dari lemak [ALM-08].

Kombinasi bahan makanan terbaik adalah bahan makanan yang memiliki jumlah kandungan gizi yang mendekati nilai dari jumlah kebutuhan gizi yang diperlukan. Selain itu, dari kombinasi tersebut akan dihasilkan biaya yang minimum. Diperlukan suatu cara agar dapat mengoptimalkan permasalahan dalam pemenuhan kebutuhan gizi dengan biaya yang minimum. Jika terdapat bahan makanan dengan biaya rendah dan kandungan gizi yang sama dengan bahan makanan dengan biaya tinggi, maka dapat dilakukan kombinasi agar dapat menekan biaya yang dikeluarkan tanpa mengurangi kandungan gizinya.

Pada penelitian [RIS-07] dan [UYU-11] menggunakan algoritma genetika untuk menentukan komposisi bahan makanan. Hasil dari kedua penelitian tersebut dalam menghasilkan komposisi bahan makanan yang mencukupi nutrisi

mendekati optimal. Namun nilai fitness yang digunakan tidak memperhitungkan harga tiap bahan makanan dan penalti apabila terjadi pelanggaran dalam mencukupi kebutuhan gizi.

Algoritma genetika banyak digunakan dalam masalah optimasi dan mempunyai kemampuan untuk menghasilkan solusi yang baik untuk masalah-masalah rumit [MAH-13]. Pada penelitian ini akan digunakan metode Genetika untuk mendapatkan hasil kombinasi kromosom terbaik dalam pemenuhan kebutuhan gizi dengan biaya minimum. Bahan makanan yang digunakan sebanyak 125, tiap bahan makanan dianggap sebagai gen. Bahan makanan tersebut masing-masing memiliki kandungan kalori, protein, lemak, karbohidrat, dan harga.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah, maka rumusan masalah yang akan diselesaikan adalah :

1. Bagaimana representasi kromosom yang dapat digunakan untuk menyelesaikan masalah optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika.
2. Bagaimana pengaruh ukuran populasi dan ukuran generasi terhadap hasil optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika.
3. Bagaimana peluang crossover dan peluang mutasi yang efektif dalam menyelesaikan permasalahan optimasi biaya pemenuhan kebutuhan energi.

1.3 Batasan Masalah

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

1. Kandungan gizi yang digunakan adalah jumlah kalori, karbohidrat, protein, dan lemak.
2. Panjang kromosom didapatkan secara random pada interval [5...15].
3. Harga yang digunakan adalah harga umum pada Kota Gresik.
4. Banyak bahan makanan yang digunakan sebanyak 125 bahan.

1.4 Tujuan

Dari identifikasi permasalahan dan batasan masalah yang ada maka tujuan penulisan skripsi ini adalah :

1. Membuat aplikasi optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika.
2. Menentukan representasi kromosom yang dapat digunakan untuk menyelesaikan masalah optimasi biaya pemenuhan kebutuhan gizi dengan algoritma genetika
3. Mengukur pengaruh ukuran populasi dan ukuran generasi terhadap hasil optimasi biaya pemenuhan kebutuhan gizi.
4. Menentukan peluang crossover dan peluang mutasi yang efektif dalam menyelesaikan permasalahan optimasi biaya pemenuhan kebutuhan gizi.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu untuk mendapatkan suatu pemecahan masalah optimasi biaya dalam memenuhi kebutuhan gizi. Sehingga dapat diketahui bahwa terdapat berbagai macam bahan makanan yang dapat digunakan untuk mencukupi kebutuhan gizi dengan biaya minimum menggunakan algoritma genetika.

1.6 Metodologi Penelitian

a. Studi Literatur

Mempelajari teori yang berhubungan dengan optimasi biaya, kebutuhan gizi, dan algoritma genetika.

b. Pendefinisian dan Analisis Masalah

Mendefinisikan dan menganalisis masalah untuk memperoleh solusi yang tepat.

c. Perancangan dan Implementasi Sistem

Membuat perancangan sistem dan mengimplementasikan hasil perancangan sistem tersebut.

d. Uji Coba dan Analisis Hasil Implementasi

Menguji coba sistem yang dihasilkan dan menganalisis hasil dari implementasi sistem tersebut apakah telah sesuai dengan tujuan yang dirumuskan sebelumnya untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

Berikut merupakan sistematika penulisan tugas akhir “Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi” dan diuraikan sebagai berikut :

BAB I Pendahuluan

Menguraikan latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari penelitian.

BAB II Tinjauan Pustaka

Menguraikan tentang dasar teori dan referensi yang mendasari proses perancangan dan implementasi pada Optimasi Biaya Pemenuhan Kebutuhan Gizi dengan Algoritma Genetika.

BAB III Metode Penelitian dan Perancangan

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi sistem Optimasi Biaya Pemenuhan Kebutuhan Gizi dengan Algoritma Genetika.

BAB IV Implementasi

Membahas proses implementasi dari perancangan sistem Optimasi Biaya Pemenuhan Kebutuhan Gizi dengan Algoritma Genetika.

BAB V Pengujian dan Analisa

Memuat hasil pengujian terhadap perangkat lunak yang telah direalisasikan. Serta menganalisa hasil pengujian tersebut.

BAB V Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam sistem Optimasi Biaya Pemenuhan Kebutuhan Gizi dengan Algoritma Genetika. Serta saran-saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma Genetika

Algoritma Genetika atau Genethic Algorithm (GA) dikenalkan oleh John Holland dalam menyelesaikan masalah optimasi. GA mensimulasikan proses yang terjadi pada populasi alamiah yang merupakan hal yang penting dalam proses evolusi. Algoritma Genetika adalah metode pencarian yang meniru perumpamaan evolusi biologis alami untuk menentukan kromosom atau individu berkualitas tinggi dalam suatu kawasan berhingga potensial yang disebut populasi. Proses pemilihan individu dari suatu populasi dievaluasi berdasarkan fungsi *fitness*. Kromosom berwujud string tersebut merupakan calon pada setiap siklus operasi yang disebut generasi [NUG-08].

Tahap evaluasi dilakukan dengan mengevaluasi kualitas setiap kromosom dalam populasi untuk menghasilkan populasi baru yang akan direkombinasi atau disilangkan kromosom berkualitas tinggi kemungkinan besar sebagai anggota *matting pool* atau induk bagi generasi berikutnya. Generasi baru inilah sebagai *subpopulasi* yang akan menggantikan posisi induk sebelumnya. Dan akan mengalami proses yang sama. Siklus ini berulang terus dan berhenti jika generasi maksimal tercapai.

Berikut ini merupakan beberapa istilah dalam GA, yaitu [SUY-07] :

1. Genome merupakan kumpulan kromosom.
2. Kromosom merupakan kumpulan dari gen.
3. Gen merupakan unit *hereditas* dan pengkodean informasi yang dapat diturunkan kepada generasi berikutnya.
4. *Allele* merupakan setting dari gen dari pengkodean.
5. *Genotype* merupakan genome yang lengkap dari suatu individu dengan semua *settingnya*.
6. *Phenotype* merupakan semua individu dengan semua sifat-sifatnya.

2.2 Struktur Umum Algoritma

Berikut merupakan struktur umum pada Algoritma Genetika menurut [KUS-03] :

1. Representasi kromosom.
2. Evaluasi dengan menghitung *fitness*.
3. Proses *crossover* untuk mendapatkan individu baru.
4. Proses mutasi yang berfungsi untuk meningkatkan variasi dalam populasi.
5. Proses seleksi untuk membentuk populasi baru.

Dari struktur ini bisa disusun siklus algoritma genetika sebagai berikut :

```
procedure AlgoritmaGenetika
begin
  t = 0
  inisialisasi P(t)
  while (bukan kondisi berhenti) do
    reproduksi C(t) dari P(t)
    evaluasi P(t) dan C(t)
    seleksi P(t+1) dari P(t) dan C(t)
    t = t + 1
  end while
end
```

Gambar 2.1 Pseudo code siklus algoritma genetika

Pada proses algoritma genetika ini diawali dengan inisialisasi, yaitu menciptakan individu-individu secara random yang mempunyai susunan gen (kromosom) tertentu. Reproduksi dilakukan untuk menghasilkan offspring dari individu-individu yang ada di populasi. Evaluasi digunakan untuk menghitung fitness setiap kromosom. Semakin besar fitness maka semakin baik kromosom tersebut untuk dijadikan calon solusi. Seleksi dilakukan untuk memilih individu dari himpunan populasi dan offSpring yang dipertahankan hidup pada generasi berikutnya [MAH-13].

2.3 Parameter-parameter Algoritma Genetika

Algoritma Genetika bekerja berdasarkan parameter-parameter tertentu yang akan mempengaruhi kinerja dan perilaku dari algoritma ini. Beberapa parameter penting yang secara langsung mempengaruhi performa dari algoritma genetika adalah ukuran populasi, jumlah generasi, probabilitas *crossover*, dan probabilitas mutasi.

2.3.1 Ukuran Populasi

Ukuran populasi menunjukkan seberapa banyak kromosom yang ada pada populasi (dalam satu generasi). Bila ada terlalu banyak kromosom, maka algoritma genetika memiliki beberapa kemungkinan untuk melakukan *crossover* dan hanya sebagian kecil *space* pencarian yang dieksplorasi. Dengan kata lain, jika ada terlalu banyak kromosom, algoritma genetika akan bergerak lambat [JUN-03].

2.3.2 Banyaknya Generasi

Generasi dapat dikatakan sebagai jumlah *iterasi* yang dilakukan terhadap proses evaluasi tiap-tiap populasi. Seperti halnya ukuran populasi, besarnya generasi akan mempengaruhi kecepatan konvergensi. Semakin besar jumlah generasi, mengakibatkan konvergensi yang lambat, tetapi bila jumlah generasi awal semakin kecil maka dapat terjadi konvergensi prematur. Untuk itu jumlah generasi yang tepat juga harus diperhitungkan dalam melakukan proses optimasi menggunakan algoritma genetika. Proses algoritma genetika akan dihentikan apabila jumlah generasi sudah terpenuhi [JUN-03].

2.3.3 Probabilitas Crossover

Probabilitas *Crossover* menunjukkan seberapa persen dari total kromosom yang akan melalui proses *crossover*. Bila tidak terjadi *crossover*, maka *offSpring* (sifat anak hasil *crossover*) merupakan salinan yang serupa dari induk (*parent*). Bila terjadi *crossover*, *offSpring* disusun dari bagian-bagian kromosom induk (*parent*). Bila probabilitas *crossover* mencapai 100%, maka semua *offSpring* disusun dari hasil *crossover*. Bila probabilitas *crossover* adalah 0%, maka keseluruhan generasi baru disusun dari salinan *genotype* populasi terdahulu yang serupa (namun hal ini tidak berarti generasi baru sama dengan generasi terdahulu). *Crossover* dilakukan dengan harapan agar tercipta sifat-sifat baru dalam *genotype* pada generasi selanjutnya yang memiliki sifat lebih baik daripada generasi induk [JUN-03].

2.3.4 Probabilitas Mutasi

Probabilitas mutasi menunjukkan seberapa sering bagian-bagian kromosom akan bermutasi. Bila tidak terjadi mutasi, *offSpring* yang akan memasuki fase seleksi alam adalah *offSpring* setelah proses *crossover* (atau disalin) tanpa adanya perubahan. Bila mutasi terjadi, bagian-bagian gen yang terpilih secara acak akan berubah. Bila probabilitas mutasi 100%, maka keseluruhan anggota di dalam populasi akan mengalami perubahan *genotype*, sedangkan bila probabilitas mutasinya yaitu 0%, maka tidak ada yang berubah [JUN-3].

2.4 Penerapan Algoritma Genetika

2.4.1 Membangun Generasi Awal

Langkah pertama dalam algoritma ini adalah membentuk sejumlah populasi awal yang digunakan untuk mencari penyelesaian optimal. Populasi awal yang dibangun dalam tugas akhir ini dengan menggunakan bilangan *random* (acak) dengan *range* bilangan yang telah ditentukan.

2.4.2 Representasi Kromosom

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Pengkodean kandidat penyelesaian ini disebut dengan kromosom. Pengkodean tersebut meliputi penyandian gen, dengan satu gen mewakili satu variabel [MAH-10].

Cara merepresentasikan permasalahan dalam kromosom merupakan suatu hal yang penting dalam algoritma genetika. Berikut contoh representasi kromosom :

Tabel 2.1 Contoh representasi kromosom

Parent	L	Kromosom														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	9	75	85	65	60	36	22	71	19	5	-	-	-	-	-	-
2	14	74	72	75	3	1	15	2	53	31	51	17	24	58	64	-
3	9	41	42	16	30	20	28	45	17	64	-	-	-	-	-	-
4	9	42	63	16	2	64	35	33	72	4	-	-	-	-	-	-

2.4.3 Operator Genetika

Operator Genetika dipergunakan untuk mengkombinasikan (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Ada dua operator genetika yaitu *crossover* dan *mutation* [MIC-96].

2.4.3.1 Persilangan (*Crossover*)

Proses perkawinan silang (*crossover*) berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk.

Secara umum, mekanisme kawin silang adalah sebagai berikut :

1. Memilih dua buah kromosom sebagai induk.
2. Memilih secara acak posisi dalam kromosom, biasa disebut titik perkawinan silang, sehingga masing-masing kromosom induk terpecah menjadi dua segmen.
3. Lakukan pertukaran antar segmen kromosom induk untuk menghasilkan kromosom anak.

Berikut merupakan contoh proses *crossover* :

Parent1	12	14	81	58	82	4	53	72	49	32	22	50	66
Parent2	9	12	54	7	70	55	7	71	82	8			
offSpring1	12	14	81	58	82	4	53	72	49	8			
offSpring2	9	12	54	7	70	55	7	71	82	32	22	50	66

Gambar 2.2 Contoh proses *crossover*

2.4.3.2 Mutasi

Operator mutasi digunakan untuk melakukan modifikasi satu atau lebih dari nilai gen dalam individu yang sama. Mutasi memastikan bahwa probabilitas untuk pencarian pada daerah tertentu dalam persoalan tidak akan pernah nol dan mencegah kehilangan total materi genetika setelah pemilihan dan penghapusan.

Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen

yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Sehingga mutasi akan meningkatkan variasi populasi [KUS-03].

Pada penelitian ini, mutasi dilakukan dengan memilih beberapa elemen pada kromosom secara acak, dan mengganti isi elemen tersebut dengan nilai *random* sebanyak jumlah bahan makanan. Berikut merupakan contoh proses mutasi :

Parent1	21	28	46	85	82	73	35	93	43
Parent2	51	2	83	32	3	48	16	23	76
off1	42	28	16	85	64	73	33	93	43
off2	44	12	83	6	3	48	16	19	76

Gambar 2.3 Contoh proses mutasi

2.4.3.3 Perhitungan *Fitness*

Nilai *fitness* adalah inversi dari biaya pelaksanaan kromosom. Nilai ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan [BAS-03]. Fungsi *fitness* yang digunakan ditunjukkan pada persamaan (2.1) :

$$Fitness = \sum TotalHargaMax - \sum Harga - Penalti \dots \dots \dots (2.1)$$

Keterangan :

$\sum TotalHargaMax$ = Total harga dari 10 bahan makanan tertinggi

$\sum Harga$ = Total harga bahan makanan pada setiap kromosom

Penalti = Penalti jika total kandungan gizi < kebutuhan gizi

Sehingga fungsi Penalti yang digunakan ditunjukkan pada persamaan (2.2) :

$$Penalti = \alpha_1(Penalti_1) + \alpha_2(Penalti_2) + \alpha_3(Penalti_3) + \alpha_4(Penalti_4) \dots (2.2)$$

Keterangan :

α_1 = Prioritas kebutuhan kalori dengan nilai 0-1

α_2 = Prioritas kebutuhan protein dengan nilai 0-1

α_3 = Prioritas kebutuhan lemak dengan nilai 0-1

α_4 = Prioritas kebutuhan karbohidrat dengan nilai 0-1

Penalti₁ = Penalti kalori

Penalti₂ = Penalti protein
 Penalti₃ = Penalti lemak
 Penalti₄ = Penalti karbohidrat

Fungsi Penalti_{1...4} yang digunakan ditunjukkan pada persamaan (2.3) :

$$Penalti_{1...4} = \begin{cases} 0, & TotalGizi \geq KebGizi \\ KebGizi - TotalGizi, & TotalGizi < KebGizi \end{cases} \dots\dots\dots(2.3)$$

Keterangan :

TotalGizi = Total kandungan gizi pada bahan makanan
 KebGizi = Kebutuhan gizi yang diperlukan

2.4.4 Seleksi

Proses seleksi bertanggung jawab untuk melakukan pemilihan terhadap individu yang hendak diikutkan dalam proses reproduksi. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Seleksi mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Ada beberapa metode seleksi dari induk, antara lain seleksi *roulette wheel*, seleksi rangking (*elitis*), dan seleksi turnamen [MAH-10].

2.4.4.1 Seleksi Elitis

Menurut [WAT-11] metode seleksi *Elitis* adalah metode dimana individu-individu yang terpilih untuk menjadi generasi selanjutnya berdasarkan pada nilai *fitness* tertinggi. Individu tersebut akan dipertahankan untuk dibandingkan dengan individu hasil proses regenerasi. Proses seleksi dilakukan secara *random* sehingga tidak ada jaminan bahwa suatu individu yang bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*nya menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa salinannya. Prosedur tersebut dikenal sebagai *elitis*.

2.5 Gizi

Istilah Gizi di Indonesia baru mulai dikenal sekitar tahun 1952-1955 sebagai terjemahan kata bahasa Inggris *nutrition*. Kata gizi berasal dari bahasa Arab “ghidza” yang berarti makanan. Menurut dialek Mesir, ghidza dibaca ghizi. Selain itu sebagian orang menerjemahkan *nutrition* dengan mengejanya sebagai nutrisi. Perkembangan ilmu gizi dan ilmu teknologi pangan mengikuti perkembangan masalah yang dihadapi manusia. Dari waktu ke waktu ilmu gizi menghadapi tantangan untuk dapat menentukan jenis dan kecukupan gizi yang optimal untuk mendukung kelangsungan hidup manusia yang makin panjang dan produktif. WHO menyatakan bahwa gizi adalah pilar utama dari kesehatan dan kesejahteraan sepanjang siklus kehidupan [SOE-00]. Unsur-unsur gizi yang terdapat dalam makanan manusia dapat memberikan energi bagi tubuh manusia yaitu karbohidrat (hidrat arang), lemak (lipid), dan protein [MOE-98].

2.5.1 Energi

Energi Basal adalah energi minimal yang diperlukan tubuh dalam keadaan istirahat, paling sedikit dua belas jam setelah makan, mental dan fisik dalam keadaan istirahat total, berbaring tapi tidak tidur, dan suhu ruangan sekitar 25°C. Energi basal ini dipengaruhi oleh luas badan (ditentukan oleh faktor berat dan tinggi badan), umur, jenis kelamin, cuaca, ras, status gizi, penyakit, dan hormon. Angka energi basal yang paling tepat adalah yang didapatkan secara empiris berdasarkan berat badan.

Energi yang diperlukan untuk melakukan kegiatan fisik didapat dari data empiris masing-masing kegiatan. Pengukuran kebutuhan energi dapat pula dilakukan dengan memonitor denyut jantung waktu melakukan aktivitas. Hasil survei gizi diberbagai daerah menunjukkan bahwa meskipun dijumpai konsumsi energi di bawah kecukupan, dalam kenyataannya mereka masih sanggup melakukan pekerjaan rumah tangga dan pekerjaan lain di sektor pertanian. Berat badan mereka yang relatif tetap menggambarkan bahwa konsumsi energinya mencukupi kebutuhannya. Dalam hal ini faktor adaptasi merupakan penjelasan

sementara yang dapat dikemukakan. Penjelasan terperinci tentang mekanisme penyesuaian diri ini belum dapat diungkapkan secara ilmiah [KAR-88].

2.6 Persamaan Kebutuhan Kalori

Persamaan Harris Benedict dapat digunakan untuk menghitung *Total Energy Expenditure* (TEE) dalam memperkirakan kebutuhan kalori manusia [HAR-97]. Langkah pertama hitung Angka Metabolisme Basal (AMB) dengan persamaan (2.4a) atau (2.4b):

- Angka Metabolisme Basal untuk Laki-laki

$$\text{Laki - laki} = 66,5 + 13,7 \text{ BB} + 5 \text{ TB} - 6,8 \text{ U} \dots\dots\dots (2.4a)$$
- Angka Metabolisme Basal untuk Perempuan

$$\text{Perempuan} = 655 + 9,6 \text{ BB} + 1,8 \text{ TB} - 4,7 \text{ U} \dots\dots\dots (2.4b)$$

Keterangan :

BB = Berat Badan dalam kg
 TB = Tinggi Badan dalam cm
 U = Umur dalam tahun

Setelah didapatkan hasil perhitungan AMB, selanjutnya langkah kedua menentukan Aktivitas Fisik (AF) seperti pada tabel berikut :

Tabel 2.2 Kelompok aktivitas fisik

Kelompok Aktivitas	Jenis Kegiatan	Faktor Aktivitas
Istirahat total	Bed rest, dalam keadaan koma	1
Sangat Ringan	Kegiatan yang dilakukan dengan berdiri atau duduk, melukis, mengemudi, kegiatan laboratorium, menulis, menjahit, menyetrikan, memasak, bermain kartu, bermain musik	1,3
Ringan	Berjalan di jalan yang rata 4-4,8km/jam, pekerjaan elektronik, memahat, pekerjaan di restoran, bengkel, membersihkan rumah, mengasuh anak, golf, bermain tennis meja.	L : 1,6 P : 1,5
Sedang	Berjalan 5,6-6,4km/jam, menyangi rumput, mencangkul, membawa beban sedang ($\pm 10\text{kg}$), bersepeda, bermain ski, bermain tennis	L : 1,7 P : 1,6
Berat	Berjalan menanjak dengan membawa beban berat, menebang pohon, bermain basket, memanjat,	L : 2,1 P : 1,9

	bermain sepak bola, berenang	
Sangat Berat	Berlari marathon, berlari/mendayung dengan cepat, bekerja sangat keras yang luar biasa	L : 2,4 P : 2,2

Sumber : Bowman B A & Russel RM . 2001. Present Knowledge in Nutrition 8th edition

Langkah ketiga yaitu menghitung *Total Energy Expenditure* (TEE) dengan persamaan (2.5) :

$$TEE = AMB \times AF \dots\dots\dots(2.5)$$

Keterangan :

AMB = Angka Metabolisme Basal

AF = Aktifitas Fisik

2.7 Karbohidrat

Karbohidrat adalah senyawa kimia yang mengandung atom C, hidrogen dan oksigen, di mana proporsi H dan O pada karbohidrat adalah 2 : 1 seperti pada air [PRA-94].

Persamaan yang digunakan untuk menghitung lemak ditunjukkan pada persamaan (2.6) :

$$Karbohidrat = 60\% \times TEE \dots\dots\dots(2.6)$$

Menurut [SUS-04] fungsi karbohidrat dalam tubuh manusia sebagai berikut :

- Sebagai sumber energi utama
Sel-sel tubuh membutuhkan ketersediaan energi yang konstan (selalu tersedia), terutama dalam bentuk glukosa.
- Pengatur metabolisme lemak
karbohidrat mencegah terjadinya oksidasi lemak yang tidak sempurna. Bila energi tidak cukup tersedia dalam tubuh, maka akan terjadi peningkatan katabolisme lemak, akibatnya terjadi akumulasi keton dan keasaman pada darah (asidosis). Dalam keadaan tersebut karbohidrat berfungsi sebagai *Fat sparer*.
- Penghemat fungsi protein

Energi merupakan kebutuhan utama bagi tubuh, Bila karbohidrat tidak mencukupi, maka protein akan dirombak untuk menghasilkan panas dan energi. Padahal fungsi utama protein adalah sebagai zat pembangun dan memperbaiki jaringan. Agar protein sesuai dengan fungsinya maka kebutuhan karbohidrat harus terpenuhi dari konsumsi makanan sehari-hari.

- d. Simpanan sebagai glikogen
glikogen menyediakan energi yang siap pakai yang tersimpan dalam hati dan otot.
- e. Pengatur peristaltik usus
Sellulosa (serat) merupakan polisakarida yang tidak dapat dicerna, tetapi mempunyai fungsi yang penting bagi kesehatan yaitu mengatur peristaltik usus (memungkinkan terjadinya gerakan usus yang teratur) dan mencegah terjadinya konstipasi (sulit buang air besar).
- f. Sumber energi utama bagi otak dan syaraf

2.8 Protein

Protein adalah senyawa organik dengan berat molekul tinggi, mengandung unsur-unsur C, H, O, dan N. Beberapa protein mengandung S dan P. Protein merupakan komponen utama jaringan tubuh yaitu untuk pertumbuhan sel, penyusun struktur sel, memelihara membran sel, mengatur keseimbangan air dalam jaringan, penyusun antibody, hormon, dan enzim [PRA-94].

Persamaan yang digunakan untuk menghitung protein ditunjukkan pada persamaan (2.7) :

$$Protein = 25\% \times TEE \dots\dots\dots(2.7)$$

Menurut [SUS-O4] fungsi protein dalam tubuh manusia sebagai berikut :

- a. Pertumbuhan dan pemeliharaan jaringan tubuh
Protein sebagai zat pembangun, merupakan bahan pembangun jaringan baru sehingga sangat penting pada masa anak-anak, remaja yang sedang tumbuh, ibu hamil dan menyusui, juga orang sakit yang dalam masa penyembuhan dan orang lanjut usia. Protein sebagai pembangun struktur

tubuh terlihat dari susunan komposisi tubuh manusia, lebih kurang 20% bagian berat badan orang dewasa terdiri dari protein.

b. Sebagai pengatur

Protein membantu mengatur keluar masuknya cairan, zat gizi dan metabolit dari jaringan masuk ke saluran darah.

c. Sebagai sumber energi

Protein mengandung unsur karbon, maka protein dapat berfungsi sebagai bahan bakar sumber energi. Bila tubuh tidak menerima karbohidrat dan lemak dalam jumlah yang cukup memenuhi, maka protein akan dibakar sebagai sumber energi. Kecukupan protein 10-15% dari total suplai kalori sehingga bila 10% dari kecukupan energi = 210 Kalori maka sama dengan 52,2 gram protein.

d. Pertahanan tubuh

Pertahanan tubuh terhadap penyakit biasanya dalam bentuk antibodi, yaitu suatu protein khusus yang dapat mengenal dan menempel atau mengikat benda-benda asing yang masuk ke dalam tubuh seperti virus, bakteri dan sel-sel asing lainnya. Protein ini dapat menjaga kekebalan tubuh terhadap infeksi.

e. Penunjang mekanis

Kekuatan dan daya tahan robek kulit dan tulang disebabkan adanya kolagen yaitu suatu protein yang berbentuk bulat panjang dan mudah membentuk serabut.

f. Alat pengangkut dan alat penyimpan

Banyak molekul dengan berat molekul kecil serta beberapa ion dapat diangkut atau dipindahkan oleh protein-protein tertentu, misalnya hemoglobine mengangkut oksigen dalam eritrosit.

2.9 Lemak

Lemak berperan sebagai sumber dan cadangan energi, sumber asam lemak esensial, pelarut vitamin A, D, E, dan K, penyebab makanan mempunyai kelunakan-kekerasan (*texture*) khusus, penyebab lamanya waktu pengosongan lambung, dan sebagai lapisan lemak tubuh dibawah kulit. Asam lemak yang

menyusun lemak terdiri dari asam lemak jenuh dan tak jenuh yang mempunyai ikatan rangkap. Tubuh kita tak mampu ataupun hanya mempunyai kemampuan terbatas untuk mensintesis asam lemak yang ikatan rangkapnya dua buah atau lebih [KAR-88].

Persamaan yang digunakan untuk menghitung lemak ditunjukkan pada persamaan (2.8) :

$$\text{Lemak} = 15\% \times TEE \dots\dots\dots(2.8)$$

Menurut [SUS-04] fungsi lemak dalam tubuh manusia sebagai berikut :

- a. Penghasil energi
Lemak dapat menyumbang energi sebanyak 9 Kalori, berarti 2,25 kali lebih besar dari energi yang dihasilkan karbohidrat dan protein. Energi yang berlebih dalam tubuh akan disimpan dalam jaringan adiposa sebagai cadangan energi.
- b. Pembangun/pembentuk struktur tubuh
Cadangan lemak yang normal terdapat dibawah kulit dan disekeliling organ tubuh, berfungsi sebagai bantalan pelindung dan penunjang letak organ tubuh. Lemak dibawah kulit juga berfungsi melindungi kehilangan panas tubuh melalui kulit yang berarti mengatur suhu tubuh.
- c. Protein *sparer*
apabila kebutuhan energi tubuh dapat dipenuhi dari karbohidrat dan lemak, maka penggunaan protein dapat dihemat supaya hanya dipergunakan sesuai fungsinya sebagai zat pembangun dan memperbaiki jaringan yang rusak.
- d. Penghasil asam lemak esensial
merupakan asam lemak yang tidak dapat disediakan oleh tubuh sehingga harus tersedia dari makanan yang dikonsumsi.
- e. Pelarut vitamin
Vitamin A,D,E dan K merupakan vitamin yang larut dalam lemak.
- f. Sebagai pelumas diantara persendian
- g. Lemak lebih lama dicerna sehingga dapat mengenyangkan

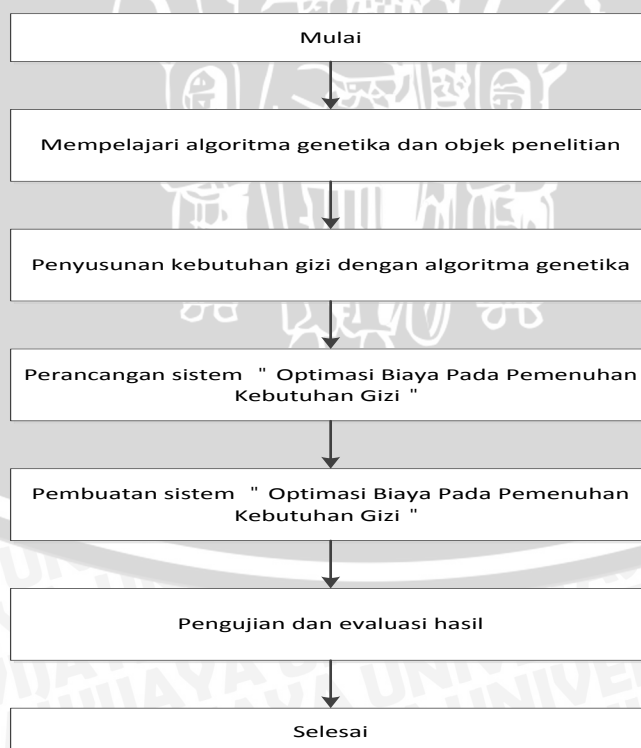
BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Tahapan Penelitian

Pada bab metodologi penelitian dan perancangan ini akan dibahas metode, rancangan yang digunakan dan langkah-langkah yang digunakan dalam pembuatan perangkat lunak dengan algoritma genetika. Tahap-tahap pembuatannya seperti gambar Gambar 3.1 adalah sebagai berikut:

1. Mempelajari metode yang digunakan pada sistem ini, yaitu algoritma genetika dan penelitian kebutuhan gizi.
2. Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
3. Membuat sistem berdasarkan analisis dan perancangan yang dilakukan.
4. Uji coba sistem dan evaluasi hasil.



Gambar 3.1 Diagram blok tahap penelitian

3.2 Analisa Kebutuhan Sistem

Analisa kebutuhan sistem merupakan tahap menganalisis hal-hal yang dibutuhkan untuk menjalankan sistem, sehingga sistem dapat berjalan secara optimal.

3.2.1 Deskripsi Umum Sistem

Pada dasarnya, kebutuhan gizi tiap manusia berbeda. Biaya juga menjadi kendala dalam membeli bahan makanan. Terkadang bahan makanan yang memiliki kandungan gizi tinggi, harganya juga mahal. Namun ada juga bahan makanan yang memiliki kandungan gizi tinggi, harganya murah. Oleh karena itu dibangun sebuah aplikasi optimasi biaya pemenuhan kebutuhan gizi dengan menerapkan algoritma genetika. Tujuannya agar biaya yang dikeluarkan dalam mendapatkan bahan makanan lebih minimum dan kandungan gizinya cukup memenuhi. Diharapkan algoritma genetika dapat menemukan solusi yang efisien karena algoritma genetika mampu membuat beberapa kemungkinan dengan lebih baik secara otomatis dan cepat.

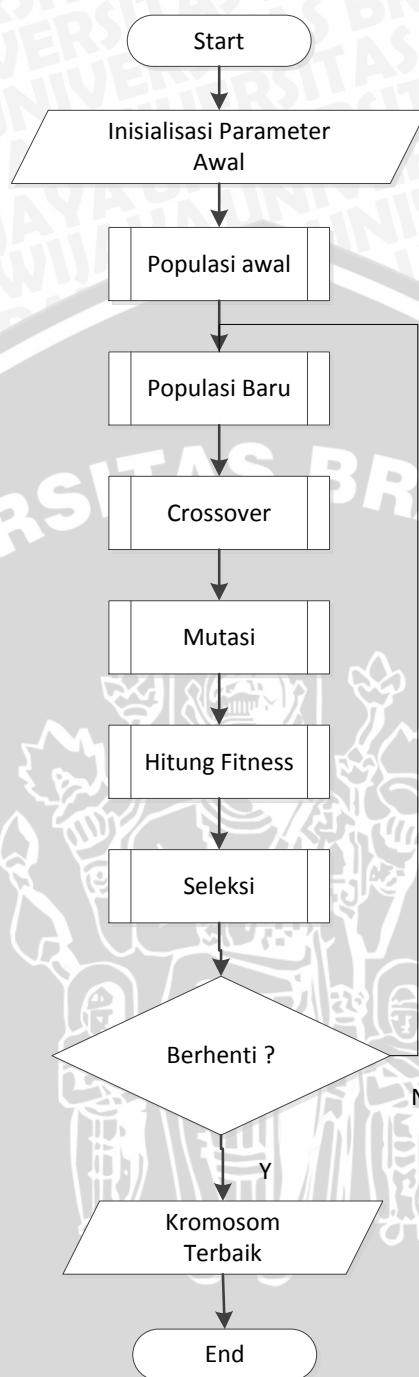
3.2.2 Data yang Digunakan

Data yang digunakan dalam penelitian adalah sebagai berikut:

1. Daftar Komposisi Bahan Makanan (DKBM) dari Depkes (2005), Subdit Gizi Klinis, Depkes Indonesia, Jakarta.
2. Daftar harga bahan makanan didapatkan berdasarkan survei di Kota Gresik pada bulan Mei 2014.

3.3 Perancangan Sistem

Pada subbab ini akan dibahas proses perancangan sistem. Algoritma Genetika mempunyai beberapa proses dalam penyelesaiannya, yaitu representasi kromosom, *crossover*, mutasi, menghitung *fitness*, dan seleksi. Berikut adalah *flowchart* proses algoritma genetika (Gambar 3.2).



Gambar 3.2 Proses algoritma genetika

3.4 Proses Algoritma Genetika

Dari gambar *flowchart* Gambar 3.2 dapat diperoleh proses dalam optimasi biaya pemenuhan kebutuhan gizi. Berikut adalah tahap-tahapnya :

1. Inisialisasi parameter awal, yaitu:
 - Memasukkan Berat Badan, Tinggi Badan, Jenis Kelamin, dan Aktivitas Fisik.
 - Memasukkan prioritas kebutuhan gizi yaitu prioritas kalori, protein, lemak, dan karbohidrat.
 - Jumlah kromosom pada setiap populasi
 - Jumlah generasi.
 - Probabilitas *crossover* (P_c)
 - Probabilitas mutasi (P_m)
2. Membangkitkan populasi awal sebanyak jumlah kromosom yang telah diinisialisasikan.
3. Membuat populasi baru dengan menggunakan proses berikut sebanyak jumlah generasi yang ditentukan :
 - Melakukan proses *crossover* pada kromosom induk yang terpilih berdasarkan p_c yang telah ditentukan.
 - Melakukan proses mutasi pada kromosom induk yang terpilih berdasarkan nilai p_m yang telah ditentukan.
 - Menghitung nilai *fitness* untuk masing-masing kromosom.
 - Melakukan proses seleksi dengan metode Elitis untuk menentukan individu pada generasi selanjutnya.
4. Apabila kondisi akhir sudah terpenuhi, maka proses selesai. Hasil akhir adalah kromosom terbaik pada generasi akhir.

Alur umum algoritma genetika pada penelitian ini mengacu pada Sub-bab

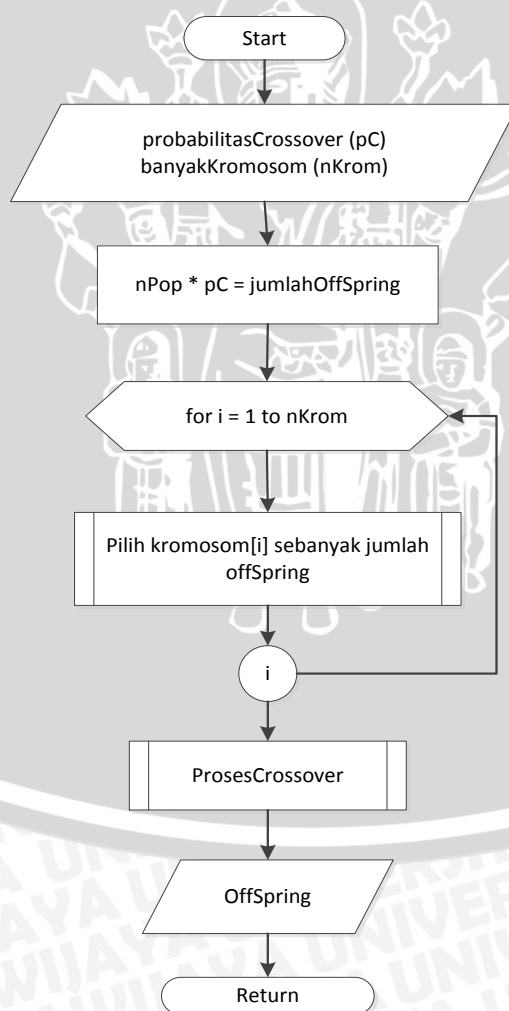
2.2. Alur tiap proses menyesuaikan permasalahan yang akan diselesaikan dan dijelaskan secara detail pada Sub-bab 3.4.1 sampai 3.4.13.

3.4.1 Membuat Populasi Awal

Pada tahap ini merupakan proses pembuatan populasi awal sebanyak jumlah populasi yang telah ditentukan. Panjang pada masing-masing kromosom berbeda, karena panjang kromosom didapatkan dari nilai *random* pada *interval* [5...15]. Tiap gen berisi nilai random sebanyak jumlah bahan makanan.

3.4.2 Crossover

Pada proses crossover akan menghasilkan *offSpring* (kromosom anak). Jumlah *offSpring* yang dibutuhkan dalam proses *crossover* dapat dilihat berdasarkan Probabilitas *Crossover* (P_c) yang telah ditentukan. Berikut merupakan flowchart dalam melakukan *crossover* (Gambar 3.3).



Gambar 3.3 Proses crossover

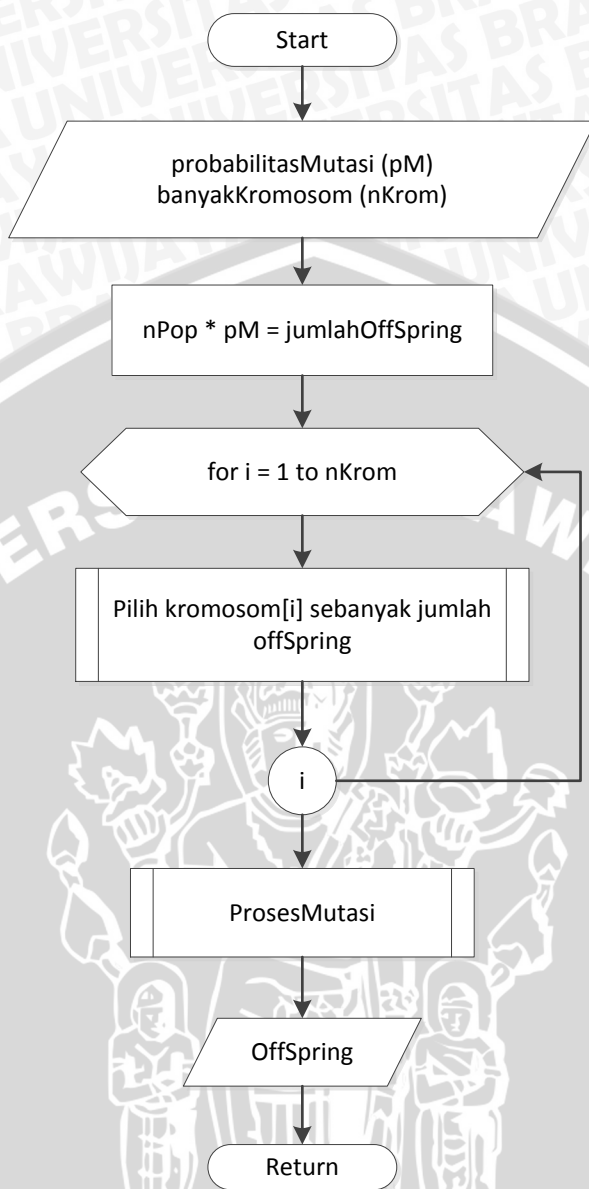
Di dalam flowchart crossover pada Gambar 3.3 terdapat subproses yaitu ProsesCrossover. Pada Gambar 3.4 adalah flowchart dari subproses ProsesCrossover.



Gambar 3.4 Detail proses crossover

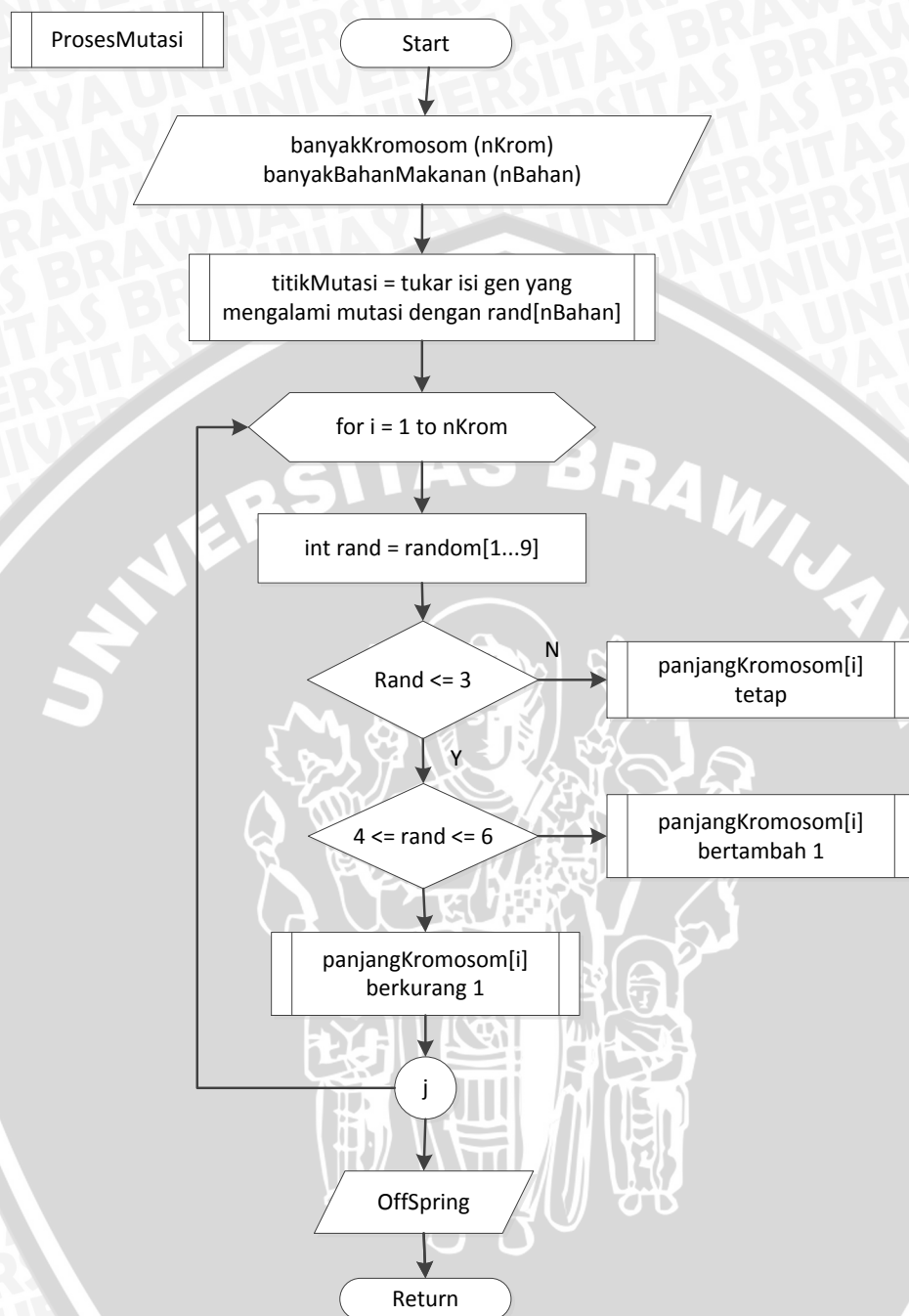
3.4.3 Mutasi

Pada proses ini sama seperti *crossover*. Proses mutasi akan menghasilkan *offSpring* (kromosom anak). Banyaknya populasi di ambil secara *random* pada tiap gen untuk dijadikan sebagai *Parent* (kromosom induk). Pada tiap kromosom yang gennya terpilih akan mengalami mutasi. Jumlah *offSpring* yang dibutuhkan dalam proses *crossover* dapat dilihat berdasarkan Probabilitas Mutasi (P_m) yang telah ditentukan. Berikut merupakan *flowchart* dalam melakukan mutasi (Gambar 3.5).



Gambar 3.5 Proses mutasi

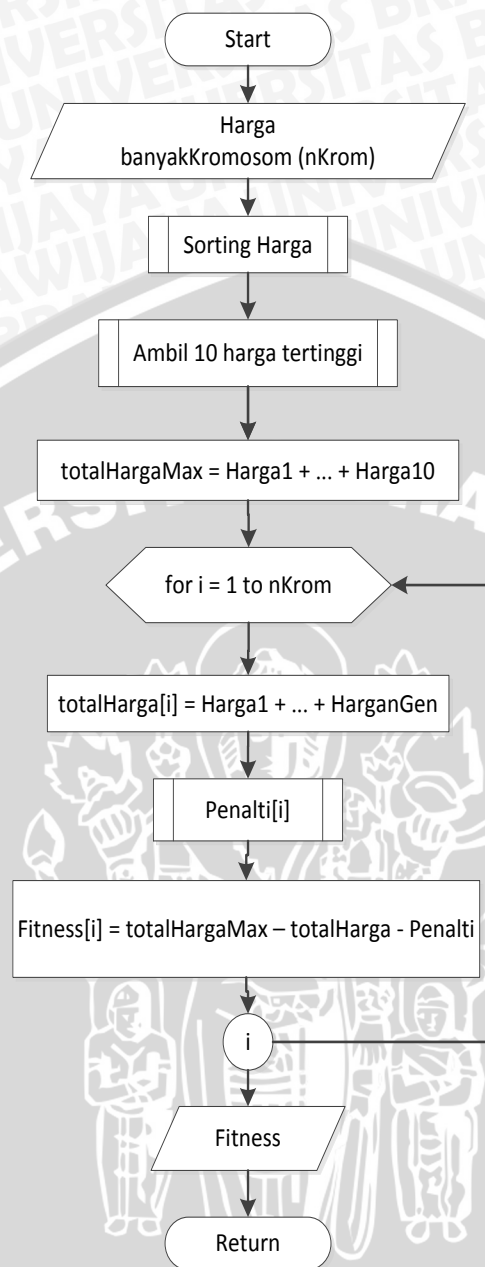
Di dalam flowchart mutasi pada Gambar 3.5 terdapat subproses yaitu ProsesMutasi. Pada Gambar 3.6 adalah flowchart dari subproses ProsesCrossover.



Gambar 3.6 Detail proses mutasi

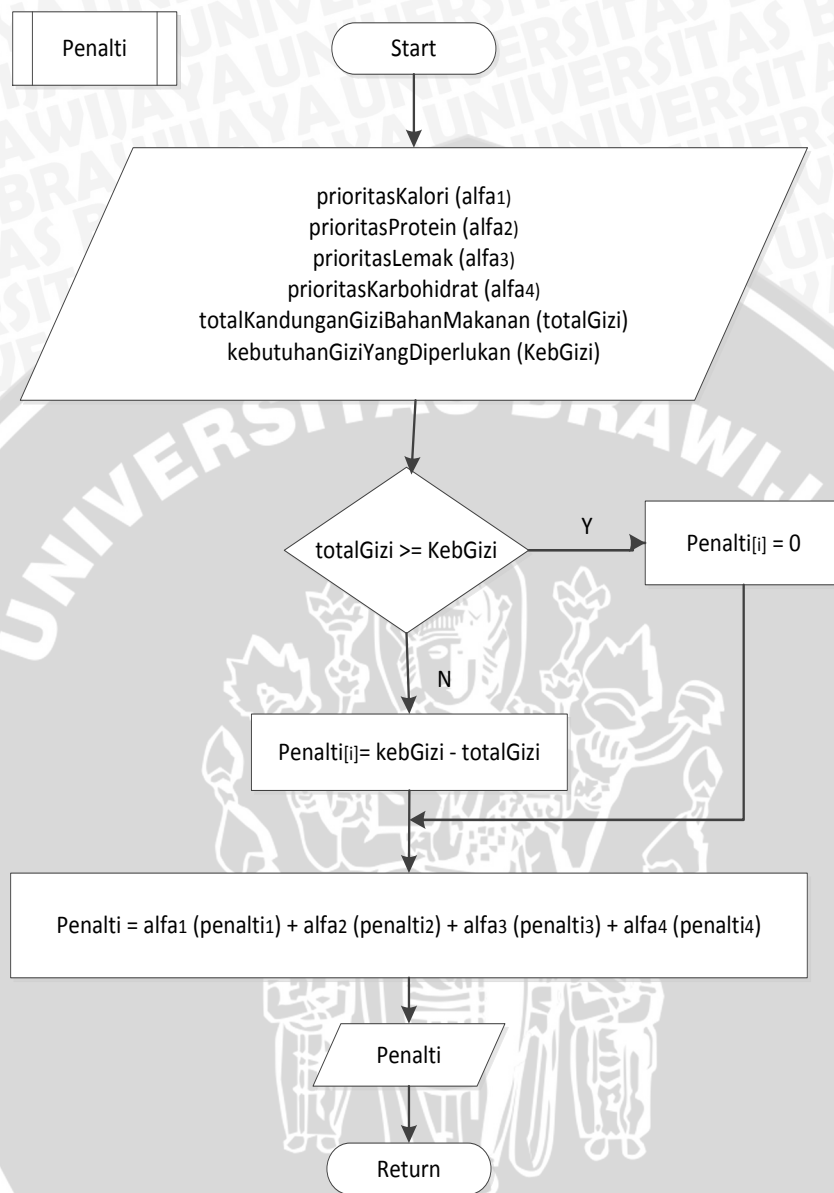
3.4.4 Menghitung Nilai Fitness

Sebelum memasuki proses seleksi, dilakukan proses perhitungan nilai *fitness* terlebih dahulu. Semua kromosom pada populasi awal dan *offSpring* hasil dari *crossover* dan mutasi dihitung nilai *fitness*-nya. Berikut *flowchart*-nya perhitungan nilai *fitness* (Gambar 3.7).



Gambar 3.7 Proses perhitungan fitness

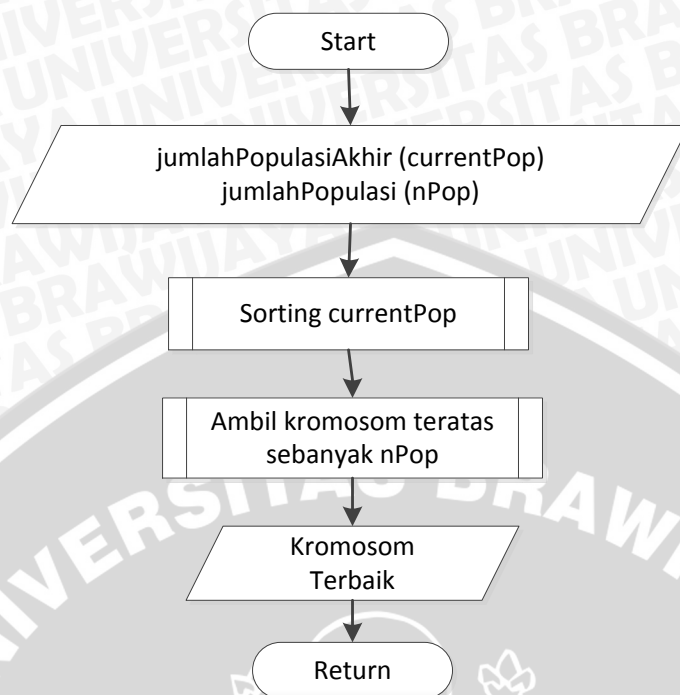
Di dalam flowchart fitness pada Gambar 3.7 terdapat subproses yaitu Penalti. Pada Gambar 3.8 adalah flowchart dari subproses Penalti. Pada proses perhitungan penalti, terdapat proses perhitungan pada tiap kandungan gizi yaitu kalori, protein, lemak, dan karbohidrat. Variabel penalti[i] adalah penalti kalori (penalti 1), penalti protein (penalti 2), penalti lemak (penalti 3), dan penalti karbohidrat (penalti 4). Keempat penalti tersebut dijadikan satu karena memiliki proses yang sama dalam perhitungannya.



Gambar 3.8 Proses penalti

3.4.5 Seleksi

Pada proses seleksi ini menggunakan metode seleksi *elitis*. Proses dalam seleksi *elitis* ini yaitu dengan melakukan pengurutan menurun berdasarkan nilai *fitness*. Selanjutnya pilih jumlah kromosom sebanyak jumlah populasi awal. Berikut adalah *flowchart* proses seleksi dengan metode *elitis* (Gambar 3.9).



Gambar 3.9 Proses seleksi elitis

3.4.6 Perhitungan Manual

Dimisalkan ada seorang perempuan dengan tinggi badan 158 cm, berat badan 48 kg, umur 22 tahun, dan Aktifitas Fisik (AF) adalah memasak. Memasak termasuk dalam kategori sangat ringan dalam Tabel 2.4 dan memiliki faktor aktifitas sebesar 1,3. Maka perhitungan kebutuhan gizinya sebagai berikut :

$$\begin{aligned}
 \text{Angka Metabolisme Basal (AMB)} &= 655 + 9,6 \text{ BB} + 1,8 \text{ TB} - 4,7 \text{ U} \\
 &= 655 + 9,6 (48) + 1,8 (158) - 4,7 (22) \\
 &= 1296,8
 \end{aligned}$$

$$\begin{aligned}
 \text{Total Energy Expenditure} &= \text{AMB} \times \text{AF} \\
 &= 1296,8 \times 1,3 = 1685,84
 \end{aligned}$$

$$\begin{aligned}
 \text{Protein} &= 25\% \times \text{TEE} \\
 &= 25\% \times 1685,84 = 421,46
 \end{aligned}$$

$$\begin{aligned}
 \text{Lemak} &= 15\% \times \text{TEE} \\
 &= 15\% \times 1685,84 = 252,876
 \end{aligned}$$

$$\begin{aligned}
 \text{Karbohidrat} &= 60\% \times \text{TEE} \\
 &= 60\% \times 1685,84 = 1011,504
 \end{aligned}$$

Pada Tabel 3.1 adalah tabel hasil kebutuhan gizi yang diperlukan.

Tabel 3.1 Kebutuhan gizi

Tabel Kebutuhan Gizi	
Kalori	1685,84
Protein	421,46
Lemak	252,876
Karbohidrat	1011,504

Setelah didapatkan hasil kebutuhan gizi, dilakukan pengurutan menurun berdasarkan harga bahan makanan. Selanjutnya pilih 10 bahan makanan teratas untuk dijumlahkan harganya. Pada tabel 3.2 adalah tabel total harga maksimal.

Tabel 3.2 Total harga maksimal

No	Nama Bahan	Kalori (kal)	Protein (g)	Lemak (g)	Karbohidrat (g)	Harga
61	Ikan Gabus	74	25,2	1,7	0	5400
19	Daging Sapi	207	18,8	14	0	4500
91	Otak	125	10,4	8,6	0,8	4500
56	Biji Jambu Monyet	562	21,6	49,6	23,6	4400
90	Hati Sapi	136	19,7	3,2	6	4000
20	Sosis Sapi	452	14,5	42,3	2,3	3500
72	Ceker Ayam	245	20,8	17,1	2,1	3500
85	Maizena	343	0,3	0	85	3500
86	Tepung Jagung	355	9,2	3,9	73,3	3500
89	Daging Domba	206	17,1	14,8	0	3500
Total Harga Maksimal						40300

3.4.7 Inisialisasi Parameter Awal

Pada tahap inisialisasi parameter awal ini yaitu menentukan banyaknya kromosom dalam satu populasi. Ukuran generasi digunakan untuk menentukan berapa banyak ukuran generasi yang ingin dihasilkan. Probabilitas *Crossover* dan Probabilitas Mutasi digunakan untuk mengetahui ukuran *offSpring* (kromosom anak) yang dihasilkan. Selain itu, menentukan banyaknya bahan makanan yang digunakan. Panjang dari masing-masing kromosom adalah nilai *random* pada interval [5...15]. Isi dari masing-masing gen adalah nilai *random* sebanyak jumlah makanan. Berikut adalah inisialisasi parameter awal :

- a. Jumlah Populasi = 20
- b. Jumlah Generasi = 1
- c. Probabilitas *Crossover* = 0,5
- d. Probabilitas Mutasi = 0,1

3.4.8 Membuat Populasi Awal

Pada proses ini adalah proses membuat populasi awal sebanyak jumlah populasi yang telah ditentukan. Proses untuk membangkitkan populasi awal yaitu suatu bahan makanan dianggap sebagai gen yang masing-masing memiliki label nomor, kalori, protein, lemak, dan karbohidrat. Pembentukan kromosomnya yaitu dengan membangkitkan nilai *random* pada *interval* [5...15] yang nantinya hasil dari nilai *random* tersebut menyatakan panjang kromosom. Selanjutnya dibangkitkan nilai *random* sebanyak jumlah bahan makanan pada tiap-tiap gen. Dimisalkan banyaknya bahan makanan adalah 125, maka nilai *random* pada *interval* [1...125]. Hasil dari populasi awal terdapat pada tabel 3.3. Pada tabel tersebut, index ke-0 menyatakan panjang kromosom. Sedangkan index ke-1 hingga indek ke-15 menyatakan isi dari tiap kromosom (gen).

Tabel 3.3 Populasi awal

Parent	Length	Kromosom														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	6	119	92	21	16	75	121	-	-	-	-	-	-	-	-	-
2	7	21	103	51	45	83	110	125	-	-	-	-	-	-	-	-
3	10	11	100	67	64	72	78	79	98	120	62	-	-	-	-	-
4	6	74	59	24	85	48	35	-	-	-	-	-	-	-	-	-
5	10	25	110	50	35	74	40	96	44	102	1	-	-	-	-	-
6	14	111	96	102	105	60	125	26	42	54	63	43	101	118	46	-
7	11	60	5	105	8	83	17	72	109	120	106	76	-	-	-	-
8	6	24	123	39	108	61	77	-	-	-	-	-	-	-	-	-
9	8	96	24	79	90	20	26	103	63	-	-	-	-	-	-	-
10	5	90	113	103	82	63	-	-	-	-	-	-	-	-	-	-
11	5	103	99	14	35	97	-	-	-	-	-	-	-	-	-	-
12	12	41	34	85	119	92	70	3	121	2	99	27	98	-	-	-
13	10	72	24	106	47	76	14	89	41	88	71	-	-	-	-	-
14	15	7	106	30	11	47	109	121	82	74	70	51	44	55	31	85
15	10	58	113	111	23	22	81	38	58	120	106	-	-	-	-	-
16	8	76	86	73	57	5	112	119	117	-	-	-	-	-	-	-
17	6	96	47	120	94	107	28	-	-	-	-	-	-	-	-	-
18	14	42	54	73	86	31	26	69	15	70	86	123	14	50	36	-
19	11	60	94	35	41	16	35	99	51	33	56	80	-	-	-	-
20	9	38	76	40	102	44	47	81	123	61	-	-	-	-	-	-

3.4.9 Menghitung Nilai *Fitness*

Untuk menghitung nilai *fitness* menggunakan Persamaan 2.1. Namun apabila kandungan gizi pada makanan < kebutuhan gizi pada Tabel 3.1, dilakukan penalti dengan menggunakan Persamaan 2.2 dan Persamaan 2.3. Pada tabel 3.4 adalah contoh perhitungan kromosom ke-1.

Parent1	119	92	21	16	75	121
---------	-----	----	----	----	----	-----

Tabel 3.4 Contoh perhitungan kromosom ke-1

No	Nama Bahan	Kalori	Protein	Lemak	Karbohidrat	Harga
119	Sawi	22	2,3	0,3	4	200
92	Pindang Layang	153	30	2,8	0	2500
21	Telur Ayam	162	12,8	11,5	0,7	1500
16	Tahu	68	7,8	4,6	1,6	2000
75	Salak	77	0,4	0	20,9	1000
121	Sari Kedelai Bubuk	344	30	20	43	1500
Total		826	83,3	39,2	70,2	8700

Setelah didapatkan total dari kandungan kalori, protein, lemak, karbohidrat, dan harga, selanjutnya total kandungan gizi tersebut dibandingkan dengan kebutuhan gizi yang terdapat pada tabel 3.1 dengan menggunakan persamaan 2.3.

$$Penalti_{1...4} = \begin{cases} 0, & TotalGizi \geq KebGizi \\ KebGizi - TotalGizi, & TotalGizi < KebGizi \end{cases}$$

Penalti Kalori = 826 < 1685,84 maka KebGizi – TotalGizi = 859,84

Penalti Protein = 83,3 < 421,46 maka KebGizi – TotalGizi = 338,16

Penalti Lemak = 39,2 < 252,876 maka KebGizi – TotalGizi = 213,676

Penalti Karbohidrat = 70,2 < 1011,504 maka KebGizi – TotalGizi = 2352,98

Kemudian setelah didapatkan penalti dari kalori, protein, lemak, dan karbohidrat, menghitung nilai penalti keseluruhan dengan persamaan 2.2.

$$Penalti = \alpha_1(Penalti_1) + \alpha_2(Penalti_2) + \alpha_3(Penalti_3) + \alpha_4(Penalti_4)$$

Dimisalkan nilai masing-masing prioritas (α) adalah 1.

$$\begin{aligned}\text{Penalti} &= 1(859,84) + 1(338,16) + 1(213,676) + 1(941,304) \\ &= 2352,98\end{aligned}$$

Setelah itu dilakukan perhitungan *fitness* dengan menggunakan Persamaan 2.1. $\sum \text{TotalHargaMax}$ dapat dilihat pada Tabel 3.2 dan $\sum \text{Harga}$ dapat dilihat pada Tabel 3.4

$$\text{Fitness} = \sum \text{TotalHargaMax} - \sum \text{Harga} - \text{Penalti}$$

$$\begin{aligned}\text{Fitness} &= 40300 - 8700 - 2352,98 \\ &= 29247,02\end{aligned}$$

Kemudian setelah dilakukan proses perhitungan *fitness*, ulangi perhitungan yang sama hingga kromosom terakhir. Berikut adalah hasil *fitness* dalam 1 populasi.

Tabel 3.5 Hasil perhitungan nilai *fitness*

Kromosom	Fitness
Parent1	29247,02
Parent2	31921,62
Parent3	24044,71
Parent4	27485,72
Parent5	28078,16
Parent6	19567,56
Parent7	20319,46
Parent8	25934,42
Parent9	23133,56
Parent10	31269,42
Parent11	34324,56
Parent12	21649,76
Parent13	19796,72
Parent14	20853,21
Parent15	26504,536
Parent16	28916,32
Parent17	31414,16
Parent18	16888,86
Parent19	20406,12
Parent20	22477,52

3.4.10 Reproduksi

Terdapat dua proses dalam melakukan reproduksi, yaitu proses *crossover* dan proses mutasi. Pada proses *crossover* dilakukan dengan menggunakan metode *One-cutpoint* sedangkan pada proses mutasi menggunakan modifikasi *reciprocal exchange*.

3.4.10.1 Crossover

One-cutpoint crossover merepresentasi kromosom untuk membangkitkan *offSpring* baru dengan mengganti sebagian gen dari induknya. Banyaknya kromosom yang dibutuhkan dalam proses *crossover* adalah $P_c \times \text{banyakKromosom} = 0,5 \times 20 = 10$. Maka dipilih 10 kromosom secara random untuk dijadikan parent pada proses *crossover*. Proses *crossover* ditunjukkan pada Gambar 3.10.

P1	119	92	21	16	75	121								
P12	41	34	85	119	92	70	3	121	2	99	27	98		
off1	119	92	21	16	92	70	3	121	2	99	27	98		
off2	41	34	85	119	75	121								

P2	21	103	51	45	83	110	125							
P11	103	99	14	35	97									
off3	21	103	51	45	97									
off4	103	99	14	35	83	110	125							

P4	74	59	24	85	48	35								
P17	96	47	120	94	107	28								
off5	74	59	24	94	107	28								
off6	96	47	120	85	48	35								

P19	60	94	35	41	16	35	99	51	33	56	80			
P19	60	94	35	41	16									
P19	60	94	35	41	16									
off8	90	113	103	82	16	35	99	51	33	56	80			

P6	111	96	102	105	60	125	26	42	54	63	43	101	118	46
P15	58	113	111	23	22	81	38	58	120	106				
off9	111	96	102	105	60	125	26	42	120	106				
off10	58	113	111	23	22	81	38	58	54	63	43	101	118	46

Gambar 3.10 Proses crossover

3.4.10.2 Mutasi

Proses *reciprocal exchange* mutation yaitu dengan memilih posisi secara random kemudian menukarkan nilai pada posisi tersebut [MAH-13]. Banyaknya kromosom yang dibutuhkan dalam proses *mutasi* adalah $P_m \times \text{banyakKromosom} = 0,5 \times 20 = 10$. Maka dipilih 10 kromosom secara random untuk dijadikan parent pada proses mutasi. Proses mutasi ditunjukkan pada Gambar 3.11.

P3	11	100	67	64	72	78	79	98	120	62
off11	11	47	67	64	72	73	79	98	120	62

P5	25	110	50	35	74	40	96	44	102	1
off12	25	110	91	22	74	40	96	37	102	1

P8	24	123	39	108	61	77
off13	119	123	39	108	5	77

P9	96	24	79	90	20	26	103	63
off14	96	24	79	90	20	26	10	63

P12	41	34	85	119	92	70	3	121	2	99	27	98
off15	41	34	85	44	92	70	3	121	2	59	27	98

P14	7	106	30	11	47	109	121	82	74	70	51	44	55	31	85
off16	7	71	30	11	47	109	121	82	99	70	51	44	89	31	85

P15	58	113	111	23	22	81	38	58	120	106
off17	58	113	111	23	22	124	38	58	120	106

P17	96	47	120	94	107	28
off18	28	47	79	94	107	28

P19	60	94	35	41	16	35	99	51	33	56	80
off19	60	94	35	41	88	35	99	12	33	56	80

P20	38	76	40	102	44	47	81	123	61
off20	38	76	40	58	40	47	81	123	61

Gambar 3.11 Proses mutasi

Selanjutnya dibangkitkan bilangan *random* pada *interval* [1...9] sebanyak jumlah *offSpring* yang mengalami mutasi. Dari hasil nilai *random* tersebut dibagi menjadi tiga proses :

1. Jika *random* pada *interval* [1...3] maka panjang kromosom tetap.
2. Jika *random* pada *interval* [4...6] maka panjang kromosom berkurang 1.
3. Jika *random* pada *interval* [7...9] maka panjang kromosom bertambah 1.

Namun apabila panjang kromosom adalah batas bawah pada *interval* dan mengalami pengurangan, maka panjang kromosom tetap. Sebaliknya apabila panjang kromosom adalah batas atas pada *interval* dan mengalami penambahan, maka panjang kromosom juga tetap seperti pada Tabel 3.6.

Tabel 3.6 Proses *random addRemove* panjang kromosom

Kromosom	Random	Panjang Kromosom
<i>offSpring11</i>	8	Add
<i>offSpring12</i>	9	Add
<i>offSpring13</i>	4	Remove
<i>offSpring14</i>	1	Tetap
<i>offSpring15</i>	8	Add
<i>offSpring16</i>	5	Remove
<i>offSpring17</i>	9	Add
<i>offSpring18</i>	2	Tetap

Proses *addRemove* pada proses mutasi dapat dilihat pada Tabel 3.7.

Tabel 3.7 Proses *addRemove* pada kromosom

offSpring	Kromosom														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<i>off11</i>	11	47	67	64	72	73	79	98	120	62	25				
<i>off12</i>	25	110	91	22	74	40	96	37	102	1	122				
<i>off13</i>	119	123	39	108	5	-									
<i>off14</i>	96	24	79	90	20	26	10	63							
<i>off15</i>	41	34	85	44	92	70	3	121	2	59	27	98	42		
<i>off16</i>	7	71	30	11	47	109	121	82	99	70	51	44	89	31	-
<i>off17</i>	58	113	111	23	22	124	38	58	120	106	69				
<i>off18</i>	28	47	79	94	107	28									
<i>off19</i>	60	94	35	41	88	35	99	12	33	56	80	81			
<i>off20</i>	38	76	40	58	40	47	81	123	61	33					

3.4.11 Evaluasi

Setelah dilakukan mutasi, selanjutnya dilakukan proses evaluasi. Proses evaluasi digunakan untuk menghitung *fitness* dan menentukan *fitness* terbaik. Semakin tinggi nilai *fitness* maka semakin baik kromosom tersebut untuk menjadi calon solusi penyelesaian masalah. Hasil *offSpring* (kromosom anak) yang dihasilkan dari proses reproduksi yaitu *crossover* dan mutasi. Kromosom *parent*

pada satu populasi akan dihitung nilai *fitness*nya. Hasil perhitungan *fitness* keseluruhan *parent* dan *offSpring* terdapat pada Tabel 3.8.

Tabel 3.8 Hasil perhitungan *fitness parent* dan *offSpring*

Kromosom	<i>Fitness</i>
Parent1	29247,02
Parent2	31921,62
Parent3	24044,71
Parent4	27485,72
Parent5	28078,16
Parent6	19567,56
Parent7	20319,46
Parent8	25934,42
Parent9	23133,56
Parent10	31269,42
Parent11	34324,56
Parent12	21649,76
Parent13	19796,72
Parent14	20853,21
Parent15	26504,536
Parent16	28916,32
Parent17	31414,16
Parent18	16888,86
Parent19	20406,12
Parent20	22477,52
offSpring1	20914,66
offSpring2	30020,12
offSpring3	34197,02
offSpring4	32137,32
offSpring5	28462,22
offSpring6	30496,66
offSpring7	29725,12
offSpring8	21535,26
offSpring9	25026,16
offSpring10	21043,736
offSpring11	21868,01
offSpring12	23035,66
offSpring13	31945,42
offSpring14	23133,56
offSpring15	20475,06
offSpring16	20300,21
offSpring17	26467,636
offSpring18	28097,92
offSpring19	16233,76
offSpring20	21712,996

3.4.12 Seleksi

Proses seleksi dilakukan untuk memperoleh hasil kromosom terbaik yang akan dijadikan populasi pada generasi berikutnya. Pada proses seleksi ini menggunakan metode *elitis*.

3.4.12.1 Elitis

Proses seleksi dengan menggunakan metode *elitis* yaitu dengan mengurutkan nilai *fitness* tertinggi. Berikut adalah tahapan seleksi metode *elitis* :

1. Urutkan semua kromosom berdasarkan nilai *fitness* tertinggi hingga terendah.
2. Ambil kromosom teratas sebanyak jumlah populasi yang telah diinisialisasikan.

Untuk contoh penerapannya dapat dilihat pada Tabel 3.9.

Tabel 3.9 Hasil seleksi elitis

Kromosom	<i>Fitness</i>
Parent11	34324,56
offSpring3	34197,02
offSpring4	32137,32
offSpring13	31945,42
Parent2	31921,62
Parent17	31414,16
Parent10	31269,42
offSpring6	30496,66
offSpring2	30020,12
offSpring7	29725,12
Parent1	29247,02
Parent16	28916,32
offSpring5	28462,22
offSpring18	28097,92
Parent5	28078,16
Parent4	27485,72
Parent15	26504,536
offSpring17	26467,636
Parent8	25934,42
offSpring9	25026,16

3.4.13 Memilih Kromosom Terbaik

Setelah dilakukan proses seleksi, selanjutnya dilakukan pemilihan kromosom terbaik berdasarkan nilai *fitness* tertinggi. Hasil kromosom terbaik pada generasi awal dapat dilihat pada Tabel 3.10.

Tabel 3.10 Hasil kromosom terbaik

Nama Kromosom	Kromosom	<i>Fitness</i>
Parent11	103 99 14 35 97	34324,56

3.5 Perancangan User Interface

Pada perancangan *User Interface* (UI) dari aplikasi ini terdiri dari dua halaman yaitu halaman utama (halaman *input*) dan halaman proses algoritma genetika.

3.5.1 Tampilan Halaman Utama

Rancangan UI halaman utama atau halaman *input* Optimasi Biaya Pemenuhan Kebutuhan Gizi dengan Algoritma Genetika dapat dilihat pada Gambar 3.12.

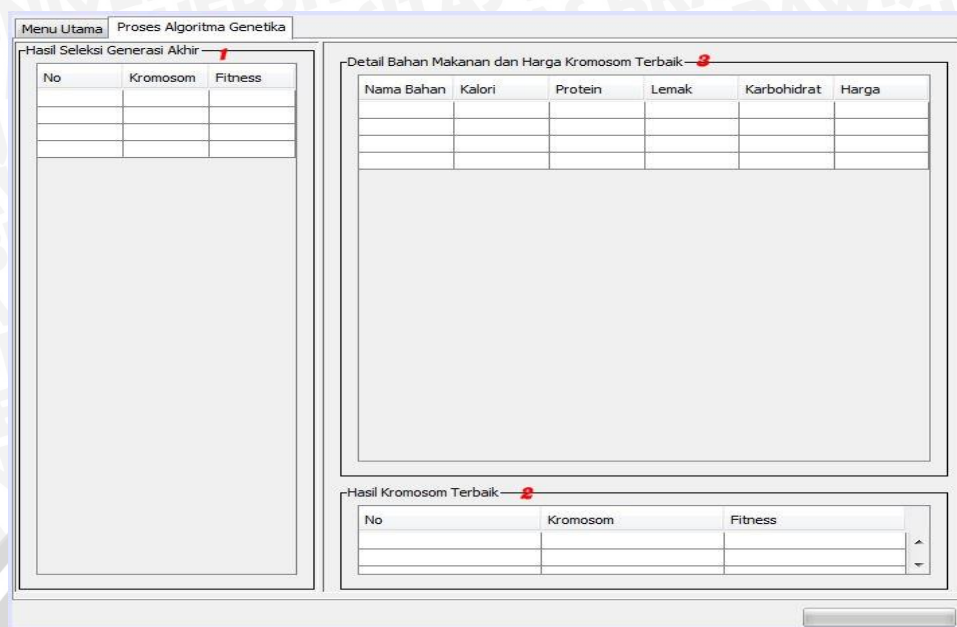
Gambar 3.12 Rancangan *user interface* halaman utama

Keterangan:

1. TextBox untuk input umur
2. TextBox untuk input berat badan
3. TextBox untuk input tinggi badan
4. RadioButton untuk input jenis kelamin
5. ComboBox untuk memilih aktifitas fisik
6. Button untuk proses perhitungan kebutuhan gizi
7. Hasil kebutuhan kalori
8. Hasil kebutuhan protein
9. Hasil kebutuhan lemak
10. Hasil kebutuhan karbohidrat
11. TextBox untuk input jumlah populasi
12. TextBox untuk input jumlah generasi
13. TextBox untuk input probabilitas *crossover*
14. TextBox untuk input probabilitas mutasi
15. TextBox untuk input prioritas kalori
16. TextBox untuk input prioritas protein
17. TextBox untuk input prioritas lemak
18. TextBox untuk input prioritas karbohidrat
19. Button untuk proses perhitungan algoritma genetika

3.5.2 Tampilan Halaman Hasil Proses Algoritma Genetika

Setelah input data pada halaman utama, hasil dari proses perhitungan dengan algoritma genetika dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan *user interface* hasil proses algoritma genetika

Keterangan:

1. Hasil seleksi dari generasi akhir
2. Hasil kromosom terbaik
3. Hasil detail bahan makanan dan harga pada kromosom terbaik

3.6 Perancangan Uji Coba dan Evaluasi

Dikarenakan tidak adanya metode yang pasti untuk menentukan parameter algoritma genetika [MAH-13], maka untuk mengevaluasi program dilakukan beberapa uji coba Antara lain:

1. Uji coba untuk menentukan ukuran populasi yang optimal untuk proses algoritma genetika pada optimasi biaya pemenuhan kebutuhan gizi .
2. Uji coba untuk menentukan ukuran generasi yang optimal untuk proses algoritma genetika pada optimasi biaya pemenuhan kebutuhan gizi.
3. Uji coba untuk mencari kombinasi probabilitas mutasi dan probabilitas *crossover* yang terbaik untuk menyelesaikan permasalahan pada optimasi biaya pemenuhan kebutuhan gizi.

3.6.1 Uji Coba Ukuran Populasi

Pada uji coba berdasarkan ukuran populasi yaitu untuk mengetahui ukuran populasi yang optimal untuk memecahkan masalah optimasi biaya pemenuhan kebutuhan gizi. Menurut [MAH-13], semakin banyak ukuran populasi berpengaruh terhadap meningkatnya kemampuan algoritma genetika dalam mencari solusi terbaik. Banyaknya populasi yang digunakan adalah kelipatan 20. Tabel yang digunakan dalam uji coba populasi dapat dilihat pada Tabel 3.11. Parameter yang digunakan pada uji coba populasi adalah :

- a. Ukuran Populasi = 20-120
- b. Ukuran Generasi = 250
- c. Probabilitas *Crossover* = 0,5
- d. Probabilitas Mutasi = 0,5

Tabel 3.11 Rancangan uji coba ukuran populasi

Banyak Populasi	Nilai <i>Fitness</i>										Rata-rata <i>Fitness</i>
	Percobaan Populasi ke -										
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											
120											

3.6.2 Uji Coba Banyaknya Generasi

Pada uji coba berdasarkan banyaknya generasi yaitu untuk mengetahui banyaknya generasi yang optimal untuk memecahkan masalah optimasi biaya pemenuhan kebutuhan gizi. Menurut [MAH-13], semakin banyak ukuran generasi berpengaruh terhadap meningkatnya kemampuan algoritma genetika dalam mencari solusi terbaik. Banyaknya jumlah generasi yang digunakan adalah kelipatan 250. Tabel yang digunakan dalam uji coba populasi dapat dilihat pada Tabel 3.12. Parameter yang digunakan pada uji coba populasi adalah :

- a. Ukuran Populasi = 20

- b. Ukuran Generasi = 250-1500
- c. Probabilitas *Crossover* = 0,5
- d. Probabilitas Mutasi = 0,5

Tabel 3.12 Rancangan uji coba banyaknya generasi

Banyak Generasi	Nilai <i>Fitness</i>										Rata- rata <i>Fitness</i>
	Percobaan Generasi ke -										
	1	2	3	4	5	6	7	8	9	10	
250											
500											
750											
1000											
1250											
1500											

3.6.3 Uji Coba Kombinasi Probabilitas *Crossover* dan Mutasi

Pada uji coba berdasarkan probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m) yaitu untuk mengetahui P_c dan P_m yang optimal untuk memecahkan masalah optimasi biaya pemenuhan kebutuhan gizi. Menurut [MAH-13], tidak ada metode pasti untuk menentukan parameter genetika yang digunakan seperti probabilitas *crossover* dan probabilitas mutasi. Kombinasi nilai probabilitas *crossover* dan probabilitas mutasi berpengaruh terhadap meningkatnya kemampuan algoritma genetika dalam mencari solusi terbaik.

Pada uji coba menggunakan algoritma genetika akan diberikan nilai-nilai yang berbeda pada parameter genetik, yaitu probabilitas *crossover* dan probabilitas mutasi. Probabilitas *crossover* dan probabilitas mutasi yang akan diujikan adalah 0 hingga 1. Tabel yang digunakan dalam uji coba kombinasi probabilitas *crossover* dan probabilitas mutasi dapat dilihat pada Tabel 3.13. Parameter yang digunakan dalam uji coba kombinasi P_c dan P_m adalah :

- a. Jumlah Populasi = Hasil populasi terbaik pada uji coba populasi
- b. Jumlah Generasi = Hasil generasi terbaik pada uji coba generasi

Tabel 3.13 Rancangan uji coba kombinasi probabilitas *crossover* dan mutasi

Kombinasi		Nilai <i>Fitness</i>										Rata-rata <i>Fitness</i>
		Percobaan Kombinasi P_c dan P_m ke -										
P_c	P_m	1	2	3	4	5	6	7	8	9	10	
1	0											
0,9	0,1											
0,8	0,2											
0,7	0,3											
0,6	0,4											
0,5	0,5											
0,4	0,6											
0,3	0,7											
0,2	0,8											
0,1	0,9											
0	1											



BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Proses implementasi adalah tahap aplikasi siap dijalankan pada keadaan yang sebenarnya. Sehingga dapat diketahui apakah sistem telah menghasilkan tujuan yang diinginkan. Terdapat dua lingkungan yang digunakan dalam implementasi, yaitu lingkungan perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam sistem optimasi biaya pemenuhan kebutuhan gizi menggunakan algoritma genetika adalah sebagai berikut:

1. Prosesor Intel(R) Core(TM) i5-450M 2,4 GHz.
2. Memori 2GB DDR3.
3. *Harddisk* dengan kapasitas 500 GB.
4. Monitor 14".

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam sistem optimasi biaya pemenuhan kebutuhan gizi menggunakan algoritma genetika adalah:

1. Sistem operasi yang digunakan adalah Windows7 *ultimate edition* 32bit.
2. *Tools* yang digunakan adalah JDK1.6 dan JRE7.
3. *Editor* yang digunakan adalah Netbeans 7.1.2.

4.2 Implementasi Program

Berdasarkan metodologi dan perancangan sistem yang terdapat pada bab 3, maka pada subbab ini akan dijelaskan implementasi proses-proses tersebut ke dalam sistem dengan menggunakan bahasa pemrograman Java.

4.2.1 Struktur Data

Struktur data yang digunakan dalam implementasi Optimasi Biaya Pemenuhan Kebutuhan Gizi Menggunakan Algoritma Genetika dapat dilihat pada *Source code 4.1*.

```
ArrayList<BahanMakanan> Bahan;
ArrayList<ArrayList<BahanMakanan>> kandidatKromosom;
```

Source code 4.1 Struktur data

Keterangan :

1. *ArrayList* Bahan = Menyimpan data bahan makanan
2. *ArrayList* kandidatKromosom = Menyimpan data kandidat kromosom

4.2.2 Membangkitkan Populasi Awal

Pada proses ini dilakukan inisialisasi kromosom sebanyak jumlah populasi awal. Jumlah populasi didapatkan dari jumlah kromosom yang diinputkan oleh *user*. Proses membangkitkan populasi awal dapat dilihat pada *Source code 4.2*

```
kandidatKromosom = new ArrayList<ArrayList<BahanMakanan>>();
for(int i=0;i<jumlahPop;i++){
    int panjangKromosom = acak.nextInt((batasAtas - batasBawah) +
    1) + batasBawah;
    ArrayList<BahanMakanan> kromosom = new ArrayList<BahanMakanan>();
    jumlahGen+=panjangKromosom;
    for(int j=0;j<panjangKromosom;j++){
        int random = acak.nextInt(Bahan.size());

        while(isInitalisasi(random)&&isInitalisasikromosom(kromosom, random)){
            random = acak.nextInt(Bahan.size());
            kromosom.add(Bahan.get(random));
            kandidatKromosom.add(kromosom);
        }
    }
}
```

Source code 4.2 Membangkitkan populasi awal

4.2.3 Perhitungan *Fitness*

Perhitungan nilai *fitness* bertujuan untuk mengetahui kualitas kromosom dalam populasi. Pada proses ini, nilai *fitness* didapatkan dari 10 harga maksimum dari harga bahan makanan, harga bahan makanan tiap kromosom dan penalti. Proses perhitungan *fitness* dapat dilihat pada *Source code 4.3*.

```
public double getFitness(ArrayList<BahanMakanan> kromosom){
    int totalHarga = 0;
    for(int i=0;i<kromosom.size();i++){
        totalHarga+=kromosom.get(i).harga;
    }
    return totalMaxHarga-totalHarga-getPinalti(kromosom);
}
```

Source code 4.3 Perhitungan fitness

4.2.4 Perhitungan Penalti

Perhitungan penalti dilakukan apabila kandungan gizi pada bahan makanan < kebutuhan gizi. Proses perhitungan penalti dapat dilihat pada *Source code 4.4*.

```
public double getPinalti(ArrayList<BahanMakanan> kromosom){
    double totalKalori = 0;
    double totalKarbohidrat = 0;
    double totalLemak = 0;
    double totalProtein = 0;
    for(int i=0;i<kromosom.size();i++){
        totalKalori+=kromosom.get(i).kalori;
        totalKarbohidrat+=kromosom.get(i).karbohidrat;
        totalProtein+=kromosom.get(i).protein;
        totalLemak+=kromosom.get(i).lemak;
    }
    double PinaltiKalori = (User.kebutuhanKalori<totalKalori)?
    User.kebutuhanKalori-totalKalori:0;
    double PinaltiLemak = (User.kebutuhanLemak<totalLemak)?
    User.kebutuhanLemak-totalLemak:0;
    double PinaltiKarbohidrat = (User.kebutuhanKarbohidrat<totalKarbohidrat)?
    User.kebutuhanKarbohidrat-totalKarbohidrat:0;
    double PinaltiProtein = (User.kebutuhanProtein<totalProtein)?
    User.kebutuhanProtein-totalProtein:0;
    return (User.bobotKalori*PinaltiKalori)+(User.bobotKarbohidrat*PinaltiKarbohidrat)+(User.bobotLemak*PinaltiLemak)+(User.bobotProtein*PinaltiProtein);
}
```

Source code 4.4 Perhitungan penalti

4.2.5 Proses Crossover

Metode *crossover* yang digunakan adalah *one-cut point*. Banyak kromosom yang mengalami *crossover* sesuai dengan nilai probabilitas *crossover* yang telah ditentukan. Proses *crossover* dapat dilihat pada *Source code 4.5*.

```
ArrayList<ArrayList<BahanMakanan>> offSpringCrossOver = new
ArrayList<ArrayList<BahanMakanan>>();
if(propCros!=0){
    int banyakDataCrossover =
    (int) (propCros*kandidatKromosom.size());
    banyakDataCrossover = (banyakDataCrossover%2==1)?
    banyakDataCrossover+1:banyakDataCrossover;
}
```

```

        ArrayList<ArrayList<BahanMakanan>>    kandidatCrossOver    =
generateKandidatCrossOver (kandidatKromosom,banyakDataCrossover);

for(int k=0;k<kandidatCrossOver.size()-1;k+=2){
    ArrayList<BahanMakanan> kromosom1 = kandidatCrossOver.get(k);
    ArrayList<BahanMakanan>          kromosom2          =
kandidatCrossOver.get(k+1);
    int posisiPotong = acak.nextInt(intervalBawahCromosom -2)+1;
    posisiPotong = ((kromosom1.size() > kromosom2.size()) ?
(kromosom2.size() - posisiPotong) : (kromosom1.size() -
posisiPotong));
    ArrayList<BahanMakanan>          kromosomBaru1      =      new
ArrayList<BahanMakanan>();
    ArrayList<BahanMakanan>          kromosomBaru2      =      new
ArrayList<BahanMakanan>();
    for (int j = 0; j < posisiPotong; j++) {
        kromosomBaru1.add(kromosom1.get(j));
        kromosomBaru2.add(kromosom2.get(j));
    }
    for (int j = posisiPotong; j < kromosom1.size(); j++) {
        kromosomBaru2.add(kromosom1.get(j));
    }
    for (int j = posisiPotong; j < kromosom2.size(); j++) {
        kromosomBaru1.add(kromosom2.get(j));
        offSpringCrossOver.add(kromosomBaru1);
        offSpringCrossOver.add(kromosomBaru2);
    }
}

```

Source code 4.5 Proses crossover

4.2.6 Proses Mutasi

Metode mutasi yang digunakan adalah modifikasi *reciprocal exchange*. Banyak kromosom yang mengalami mutasi sesuai dengan nilai probabilitas mutasi yang telah ditentukan. Proses mutasi dapat dilihat pada Source code 4.6.

```

ArrayList<ArrayList<BahanMakanan>>    offSpringMutasi    =      new
ArrayList<ArrayList<BahanMakanan>>();
if(propMutasi!=0){
    int          banyakDataMutasi
(int) (propMutasi*kandidatKromosom.size());
    ArrayList<ArrayList<BahanMakanan>>    kandidatMutasi    =
generateKandidatMutasi (kandidatKromosom,banyakDataMutasi);
    for(ArrayList<BahanMakanan> mutasiKromosom : kandidatMutasi){
        ArrayList<Integer> indexGen = new ArrayList<Integer>();
        int counter = 0;
        while(indexGen.size()<=banyakMutasiTiapGen){
            double random = Math.random();
            if(random> 0.5){
                indexGen.add(counter);
            }
            counter++;
            if(counter>=mutasiKromosom.size())
                counter=0;
            for(Integer index : indexGen ){
                int random = acak.nextInt(Bahan.size());

```

```
mutasiKromosom.set(index, Bahan.get(random));}}
offSpringMutasi=kandidatMutasi;}
```

Source code 4.6 Proses mutasi

4.2.7 Proses Seleksi Elitis

Metode seleksi yang digunakan adalah *elitis*. Pada seleksi ini semakin besar nilai *fitness* suatu individu maka akan semakin besar range individu tersebut untuk terpilih menjadi induk. Tahap seleksi *elitis* yaitu menurunkan semua kromosom berdasarkan *fitness* tertinggi hingga terendah, setelah itu ambil kromosom sebanyak jumlah populasi yang telah ditentukan. Kromosom yang terpilih akan menjadi kromosom *parent* pada generasi selanjutnya. Proses seleksi *elitis* dapat dilihat pada Source code 4.7.

```
ShortFitnes(kandidatKromosom);
ArrayList<ArrayList<BahanMakanan>> kandidatK = new
ArrayList<ArrayList<BahanMakanan>> ();
for(int j=0;j<jumlahPop;j++){
    kandidatK.add(kandidatKromosom.get(j));
    kandidatKromosom = kandidatK;}
```

Source code 4.7 Proses seleksi elitis

4.2.8 Pemilihan Kromosom Terbaik

Hasil dari kromosom terbaik didapatkan dari kromosom yang memiliki nilai *fitness* tertinggi dari generasi akhir. Proses pemilihan kromosom terbaik dapat dilihat pada Source code 4.8.

```
public void ShortFitnes(ArrayList<ArrayList<BahanMakanan>>
bahanShort){
for(int i=0;i<bahanShort.size();i++){
    for(int j=0;j<bahanShort.size();j++){
        if(getFitness(bahanShort.get(i))>getFitness(bahanShort.get(j))){
            ArrayList<BahanMakanan> a = bahanShort.get(i);
            bahanShort.set(i, bahanShort.get(j));
            bahanShort.set(j, a);
        }
    }
}
}
```

Source code 4.8 Pemilihan kromosom terbaik

4.3 Implementasi User Interface

User Interface pada sistem terdiri dari 2 halaman yaitu halaman utama dan halaman proses algoritma genetika. Pada halaman awal berfungsi untuk menjalankan proses kebutuhan gizi dan input parameter untuk proses algoritma

genetika. Pada Gambar 4.1 merupakan implementasi *user interface* pada halaman utama.

Input Data User

Usia (thn) : 22
 Berat Badan (kg) : 48
 Tinggi Badan (cm) : 158
 Jenis Kelamin : ☐ Laki-laki ☒ Perempuan
 Aktivitas Fisik : Sangat Ringan
 Kegiatan yang dilakukan dengan berdiri atau duduk, melukis, mengemudi, kegiatan laboratorium, menulis, menjahit, menyetraka, memasak, bermain kartu, bermain musik.

Hasil Kebutuhan Gizi

Kebutuhan Kalori : 1685.84
 Kebutuhan Protein : 421.46
 Kebutuhan Lemak : 252.87599999999998
 Kebutuhan Karbohidrat : 1011.5039999999999

Input Data Algen

Jumlah Populasi : 20
 Jumlah Generasi : 1
 Probabilitas Crossover : 0.5
 Probabilitas Mutasi : 0.5

Prioritas Gizi

Prioritas Kalori : 1.0
 Prioritas Protein : 1.0
 Prioritas Lemak : 1.0
 Prioritas Karbohidrat : 1.0

Buttons: Kebutuhan Gizi, Proses Algen

Gambar 4.1 Implementasi user interface halaman utama

Pada halaman kedua berfungsi untuk menampilkan hasil dari proses algoritma genetika yaitu hasil seleksi generasi akhir, hasil kromosom terbaik, dan detail bahan makanan dan harga dari kromosom terbaik. Pada Gambar 4.2 merupakan implementasi *user interface* hasil proses algoritma genetika.

Hasil Seleksi Generasi Akhir

No	Nama	Kromosom	Fitness
31	offspring ke-11	90 13 102 71 87	28900.0
8	Parent ke-8	76 51 1 33 101 16 87 79 64	28900.0
9	Parent ke-9	92 100 104 13 14 89 123	28200.0
36	offspring ke-16	90 51 101 7 101 69 20 76 64	28040.0
10	Parent ke-10	35 122 28 6 66 62 116 42 14 68 112 5 96 1	27802.16
34	offspring ke-14	115 66 64 96 118 86 106 23 124 91 97 59	27320.16
11	Parent ke-11	86 9 99 91 108 18 98 33 15 117 63 8	25097.16
12	Parent ke-12	67 70 85 74 116 32 84 82	24350.0
27	offspring ke-7	106 104 61 112 70 67 40 17 83 8 102 66 86	23728.16
26	offspring ke-6	15 4 27 45 68 22 30 42 91 79 119	23509.16
13	Parent ke-13	8 58 102 89 12 59 20 77 28 37 119	23390.16
14	Parent ke-14	8 121 64 71 97 108 123 78 35 119 72 77 39 63	22885.16
15	Parent ke-15	15 4 27 45 68 22 30 42 91 79 82	22716.16
25	offspring ke-5	8 58 102 89 12 59 20 77 28 37 82	22597.16
38	offspring ke-18	32 76 28 67 66 60 116 42 14 68 124 10 96 1	20554.16
16	Parent ke-16	36 121 110 111 52 69 55 80 76 51 92 86 78 49	20067.16
17	Parent ke-17	77 106 68 88 118 26 34 91 1 70 72 97 78	19634.16
18	Parent ke-18	2 46 67 63 16 7 29 34 43 104 10 100 85 53 73	19418.16
40	offspring ke-20	4 121 84 71 7 108 108 78 35 119 72 79 39 24	19027.16
19	Parent ke-19	106 104 61 112 70 67 40 17 83 8 72 27 60 10...	16411.16
20	Parent ke-20	82 4 17 70 90 3 69 85 77 89 102 66 86	15956.16
28	offspring ke-8	82 4 17 70 90 3 69 85 77 89 72 27 60 101 120	8639.16

Detail Bahan Makanan dan Harga Kromosom Terbaik

Nama Bahan	Kalori	Protein	Lemak	Karbohidrat	Harga
Daun Pepaya	79.0	8.0	2.0	11.9	1000
Bawang Bombay	45.0	1.4	0.2	10.3	300
Kacang Panjang	357.0	17.3	1.5	70.0	200
Anggur	50.0	0.5	0.2	12.8	2300
Bawang Putih	95.0	4.5	0.2	23.1	700

Hasil Kromosom Terbaik

No	Nama	Kromosom	Fitness
30	offspring ke-10	38 33 101 78 35	35800.0

Gambar 4.2 Implementasi user interface hasil proses algoritma genetika

BAB V

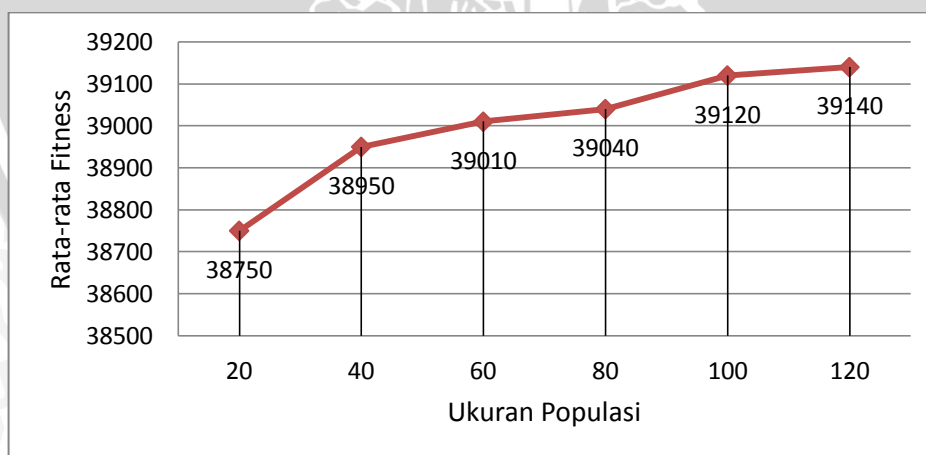
PENGUJIAN DAN ANALISA

5.1 Pengujian dan Analisa Ukuran Populasi

Pada pengujian ini bertujuan untuk mengetahui ukuran populasi yang optimal pada permasalahan optimasi biaya kebutuhan gizi. Banyak populasi yang digunakan adalah kelipatan 20. Ukuran generasi yang digunakan yaitu 250 generasi. Sedangkan kombinasi P_c dan P_m yang digunakan adalah 0,5 : 0,5. Pengujian dilakukan masing-masing 10 kali. Dari 10 percobaan tersebut akan dirata-rata untuk mengetahui populasi yang optimal dari masing-masing populasi. Pada Tabel 5.1 merupakan tabel percobaan ukuran populasi.

Tabel 5.1 Hasil uji coba ukuran populasi

Banyak Populasi	Nilai Fitness										Rata-rata Fitness
	Percobaan Populasi ke -										
	1	2	3	4	5	6	7	8	9	10	
20	38800	38700	38800	38700	38700	38800	38600	39000	38500	38900	38750
40	39100	38900	39200	38700	38900	38600	38900	39100	39000	39100	38950
60	39200	38800	39000	39000	38800	39100	39200	39000	38900	39100	39010
80	39300	39200	39000	39000	38800	38900	39300	38800	39100	39000	39040
100	39100	39200	39100	39200	39100	39300	39200	39100	39100	38800	39120
120	39200	39000	39000	39100	39000	39200	39300	39200	39200	39200	39140



Gambar 5.1 Grafik hasil uji coba ukuran populasi

Pada grafik Gambar 5.1 dapat dilihat kenaikan rata-rata *fitness* untuk 10 kali percobaan dari 20 populasi hingga 120 populasi. Dari grafik tersebut kenaikan rata-rata *fitness* terbaik pada populasi 40 yaitu 38950. Hal ini dapat

disimpulkan bahwa populasi 40 merupakan ukuran populasi yang optimal untuk menyelesaikan masalah optimasi kebutuhan gizi. Apabila ukuran populasi semakin tinggi maka akan berpengaruh pada rata-rata *fitness*. Namun pada populasi 40 merupakan titik optimum karena tidak terjadi lagi kenaikan rata-rata *fitness* yang signifikan setelah populasi diatas 40.

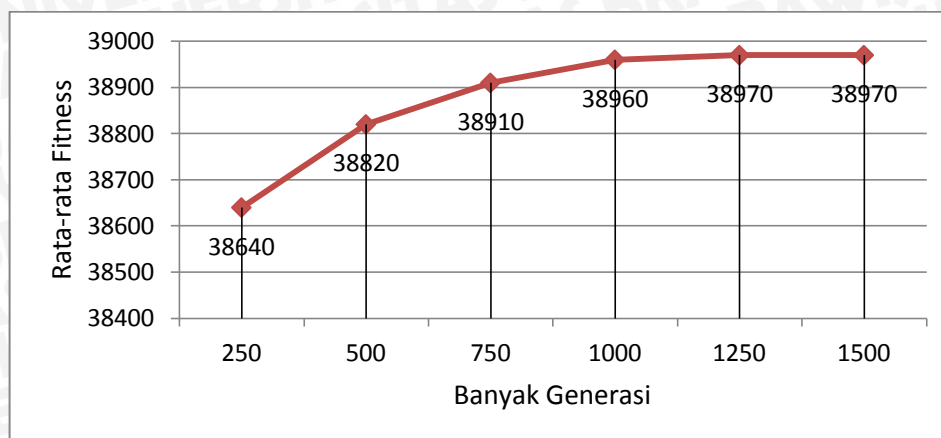
Percobaan ukuran populasi hanya dilakukan sampai populasi ke 120 karena nilai rata-rata *fitness* setelah populasi ke 120 sama, yaitu 39140. Hal tersebut bisa dikatakan konvergen, karena tidak ada kenaikan nilai rata-rata *fitness* pada populasi berikutnya.

5.2 Pengujian dan Analisa Banyaknya Generasi

Pada pengujian ini bertujuan untuk mengetahui ukuran generasi yang optimal pada permasalahan optimasi biaya kebutuhan gizi. Ukuran generasi yang digunakan adalah kelipatan 250. Ukuran populasi yang digunakan yaitu 20 populasi. Sedangkan kombinasi P_c dan P_m yang digunakan adalah 0,5 : 0,5. Pengujian dilakukan masing-masing 10 kali. Dari 10 percobaan tersebut akan dirata-rata untuk mengetahui generasi yang optimal dari masing-masing generasi. Pada Tabel 5.2 merupakan tabel percobaan banyaknya generasi.

Tabel 5.2 Hasil uji coba banyaknya generasi

Banyak Generasi	Nilai Fitness										Rata- rata Fitness
	Percobaan Generasi ke -										
	1	2	3	4	5	6	7	8	9	10	
250	38700	38600	38600	38400	39100	38700	38500	38800	38700	38300	38640
500	38900	39100	38900	38700	38900	38700	39000	38600	38900	38500	38820
750	39000	38700	38900	39200	38900	39100	38800	38700	38800	39000	38910
1000	39000	39000	39300	39100	38900	38900	38600	39000	38900	38900	38960
1250	39000	39100	38900	39100	39100	39000	38900	38900	38800	38900	38970
1500	39000	39100	38800	39000	38800	38800	39200	38900	38800	39300	38970



Gambar 5.2 Grafik hasil uji coba banyaknya generasi

Pada grafik Gambar 5.2 dapat dilihat kenaikan rata-rata *fitness* untuk 10 kali percobaan dari 250 generasi hingga 1500 populasi. Dari grafik tersebut kenaikan rata-rata *fitness* terbaik pada generasi 500 yaitu 38820. Hal ini dapat disimpulkan bahwa generasi 500 merupakan jumlah generasi yang optimal untuk menyelesaikan masalah optimasi kebutuhan gizi. Apabila jumlah generasi semakin tinggi maka akan berpengaruh pada rata-rata *fitness*. Semakin tinggi jumlah generasi belum tentu menghasilkan nilai yang optimal. Selain itu hal tersebut akan membutuhkan waktu lama untuk prosesnya. Pada generasi 500 merupakan titik optimum karena tidak terjadi lagi kenaikan rata-rata *fitness* yang signifikan setelah generasi diatas 250.

Percobaan ukuran generasi hanya dilakukan sampai generasi ke 1500 karena nilai rata-rata *fitness* setelah populasi ke 1500 sama, yaitu 38960. Hal tersebut bisa dikatakan konvergen, karena tidak ada kenaikan nilai rata-rata *fitness* pada generasi berikutnya.

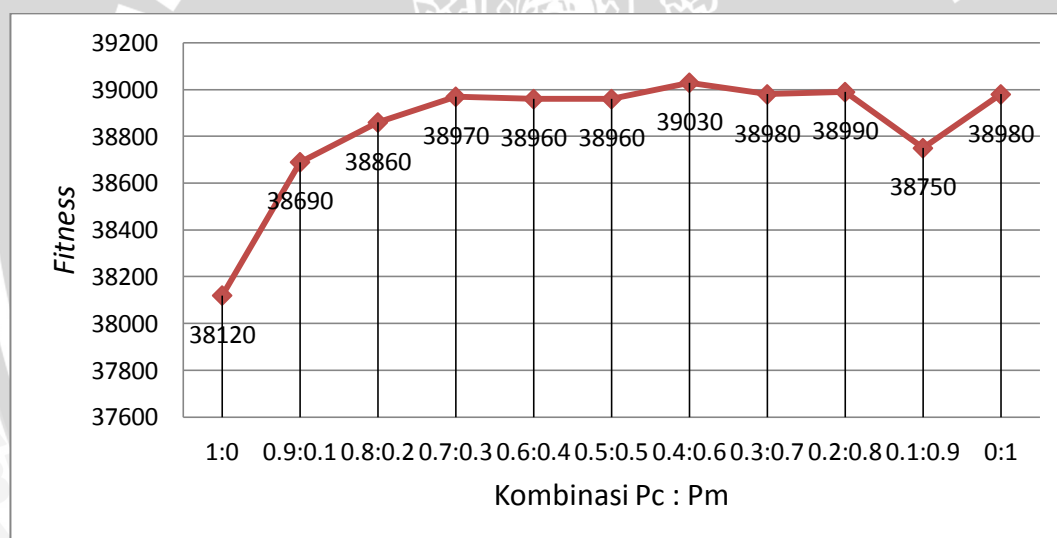
5.3 Pengujian dan Analisa Kombinasi Probabilitas Crossover dan Probabilitas Mutasi

Pada pengujian ini bertujuan untuk mengetahui kombinasi probabilitas *crossover* dan probabilitas mutasi yang optimal pada permasalahan optimasi biaya kebutuhan gizi. Banyak populasi dan generasi yang digunakan adalah populasi dan generasi terbaik pada uji coba populasi dan generasi. Sedangkan kombinasi yang digunakan yaitu nilai 0 hingga 1. Pengujian dilakukan masing-masing 10

kali. Dari 10 percobaan tersebut nilai fitness akan dirata-rata untuk mengetahui kombinasi P_c dan P_m yang optimal dari masing-masing kombinasi. Pada Tabel 5.3 merupakan tabel percobaan kombinasi P_c dan P_m .

Tabel 5.3 Hasil uji coba kombinasi probabilitas *crossover* dan probabilitas mutasi

Kombinasi		Nilai Fitness										Rata-rata Fitness
		Percobaan Kombinasi P_c dan P_m ke -										
Pc	Pm	1	2	3	4	5	6	7	8	9	10	
1	0	38400	38200	38500	37900	37600	37300	38400	38400	38300	38200	38120
0,9	0,1	38700	38600	38900	38400	38500	38300	38800	38900	39000	38800	38690
0,8	0,2	38800	38800	38600	39200	38900	38500	39100	38800	39000	38900	38860
0,7	0,3	39000	39000	39000	39100	39100	39100	38800	39000	38800	38800	38970
0,6	0,4	38800	39000	39000	39100	39100	39000	38600	38800	39100	39100	38960
0,5	0,5	39000	38900	39000	39000	38900	39100	39000	38700	38800	39200	38960
0,4	0,6	38800	39100	38900	39100	39000	39000	39300	39100	39000	39000	39030
0,3	0,7	38900	39100	39000	39200	38900	39300	38800	38900	39000	38700	38980
0,2	0,8	39100	38800	39300	38900	39000	39200	38700	39100	39100	38700	38990
0,1	0,9	38600	39100	38700	38600	38700	38900	39000	38700	38700	38500	38750
0	1	38900	38900	39100	38800	39000	38800	39200	39000	39100	39000	38980



Gambar 5.3 Grafik hasil uji coba kombinasi probabilitas *crossover* dan probabilitas mutasi

Pada grafik Gambar 5.3 dapat dilihat rata-rata *fitness* terbaik dan optimal pada uji coba ini adalah 39030 yaitu pada kombinasi probabilitas *crossover* 0,4 dan probabilitas mutasi 0,6. Kombinasi terburuk yaitu pada kombinasi probabilitas *crossover* 1 dan probabilitas mutasi 0 dengan rata-rata *fitness* 38120. Maka dapat disimpulkan kombinasi $P_c : P_m$ terbaik adalah 0,4:0,6. Apabila menggunakan nilai P_c yang rendah dan nilai P_m tinggi maka algoritma genetika akan bekerja seperti *random search* dan tidak mampu untuk mengeksplorasi

daerah pencarian secara efektif. Pada kondisi sebaliknya, apabila nilai P_c tinggi dan P_m rendah maka algoritma genetika tidak akan mampu memperlebar area pencarian [MAM-13].



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil uji coba dalam skripsi ini adalah sebagai berikut :

1. Bentuk representasi kromosom yang digunakan memiliki panjang kromosom pada interval $[5 \dots 15]$ yang didapatkan secara random. Dari adanya panjang kromosom yang berbeda-beda, *offSpring* yang dihasilkan akan lebih bervariasi sehingga dapat mempengaruhi hasil *fitness*.
2. Dari hasil uji coba dapat disimpulkan bahwa ukuran populasi dan ukuran generasi sangat berpengaruh terhadap hasil. Pada uji coba ukuran populasi, populasi optimal adalah 40 populasi dengan rata-rata *fitness* 38950. Pada uji coba banyaknya generasi, generasi optimal adalah 500 generasi dengan rata-rata *fitness* 38820.
3. Peluang *crossover* dan mutasi dilakukan dengan uji coba kombinasi. Hasil yang diperoleh dari uji coba kombinasi probabilitas *crossover* dan probabilitas mutasi yang terbaik adalah probabilitas *crossover* 0,4 dan probabilitas mutasi 0,6 dengan rata-rata *fitness* 39030.

6.2 Saran

Berdasarkan hasil kesimpulan, titik optimal pada uji populasi dan generasi relatif rendah. Hal tersebut dapat terjadi karena pengaruh dari jumlah data, metode *crossover* dan metode mutasi yang digunakan. Selain itu probabilitas *crossover* dan mutasi yang digunakan pada saat uji populasi dan generasi adalah 0,5:0,5 (bukan kombinasi yang efektif pada hasil uji coba $P_c P_m$). Pada permasalahan yang lebih kompleks ini, hibridisasi dapat digunakan untuk memperkuat kemampuan algoritma genetika [MAH-14].

DAFTAR PUSTAKA

- [ALM-08] Almatsier, S. 2008. *Penuntun Diet*. Gramedia Pustaka Utama : Jakarta.
- [BAS-03] Basuki, Achmad. 2003. *Algoritma Genetika Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*. Skripsi PENS-ITS : Surabaya
- [HAR-97] Hartono, Andry. 1997. *Asuhan Nutrisi Rumah Sakit*. Penerbit Buku Kedokteran EGC : Jakarta.
- [JUN-03] Juniawati. 2003. *Implementasi Algoritma Genetika Untuk Mencari Volume Terbesar Bangun Kotak Tanpa Tutup Dari Suatu Bidang Datar Segi Empat*. Jurnal Ilmiah LPPM Universitas Surabaya.
- [KAR-88] Karyadi. 1988. *Kecukupan Gizi yang Dianjurkan*. Gramedia : Jakarta.
- [KUS-03] Kusumadewi, Sri. 2003. *Artificial Intelligence Teknik dan Aplikasinya*. Graha Ilmu : Yogyakarta.
- [MHA-92] Mahan. 1992. *Food, Nutrition, and Diet Therapy*. W.B Saunders Company : USA.
- [MAH-10] Mahmudy, WF dan Wahyu, AW. 2010. *Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner*. Jurnal Ilmiah Kursor 5:1.
- [MAH-13] Mahmudy, Wayan Firdaus. 2013. *Algoritma Evolusi*. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya : Malang.
- [MAH-14] Mahmudy, WF, Marian, RM dan Luong, LHS. 2014. *Hybrid Genetic Algorithms For Part Type Selection and Machine Loading Problems With Alternative Production Plans In Flexible Manufacturing System*. ECTI Transactions on Computer and Information Technology (ECTI-CIT), Vol.8 No.1, PP.80-93.
- [MAM-13] Mahmudy, WF, Marlian, RM dan Luong, LHS. 2013. *Real Coded Genetic Algorithms For Solving Flexible Job Shop Scheduling*

- Problem – Part II : Optimization. Advanced Materials Research, Vol 701, pp. 364-369.*
- [MIC-96] Michalewics, Z. 1996. *Genetic Algorithm + Data Structure = Evolution Programs*. Springer-Verlag 3rd Edition : New York.
- [MOE-98] Moehji, Sjahmien. *Ilmu Gizi Jilid 3*. Bhratara Karya Aksara : Jakarta.
- [NUG-08] Nugraha, Ivan. 2008. *Algoritma Genetika Untuk Optimasi Penjadwalan Kegiatan Belajar Mengajar*. Makalah IF2251 Strategi Algoritmik 1.
- [PRA-94] Prawirokusumo, Soeharto. 1994. *Ilmu Gizi Komparatif*. BPFE : Yogyakarta.
- [RIS-07] Rismawan, Tedy. 2007. *Aplikasi Algoritma Genetika Untuk Penentuan Komposisi Bahan Pangan Harian*. Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007). Universitas Islam Indonesia : Yogyakarta.
- [SOE-00] Soekirman. 2000. *Ilmu Gizi dan Aplikasinya*. Departemen Pendidikan Nasional : Jakarta.
- [SUS-04] Susanto, Tri. 2004. *Dasar-dasar ilmu pangan dan gizi*. Akademika : Yogyakarta
- [SUY-07] Suyanto. 2007. *Artificial Intelligence Searching, Reasoning, Planning dan Learning*. Informatika : Bandung.
- [TOR-09] Toriq, Adib. 2009. *Optimasi Formulasi Pakan Ternak Dengan Menggunakan Metode Genetika*. Skripsi Universitas Brawijaya : Malang.
- [UYU-11] Uyun, Shofwatul. 2011. *Penentuan Komposisi Bahan Pangan Untuk Diet Penyakit Ginjal Dan Saluran Kemih Dengan Algoritma Genetika*. Seminar Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011). Universitas Islam Negeri Sunan Kalijaga : Yogyakarta.
- [WAT-11] Wati, Anastasia Widya. 2011. *Penerapan Algoritma Genetika Dalam Optimasi Model Dan Simulasi Dari Suatu Sistem*. Jurnal Keilmuan Teknik Industri 1:4.

LAMPIRAN DATA HARGA DAN BAHAN MAKANAN

No	Nama Bahan	Kalori (kal)	Protein (g)	Lemak (g)	Karbohidrat (g)	Harga
1	Beras	360	6,8	0,7	78,9	900
2	Beras Ketan Putih	362	6,7	0,7	79,4	1200
3	Beras Ketan Hitam	356	7	0,7	78	960
4	Bihun	360	4,7	0,1	82,1	800
5	Biskuit	458	6,9	14,4	75,1	400
6	Kentang	83	2	0,1	19,1	700
7	Makaroni	363	8,7	0,4	78,7	600
8	Maizena	343	0,3	0	85	1060
9	Tapioka	362	0,5	0,3	86,9	700
10	Tepung Beras	364	7	0,5	80	1000
11	Tepung Sagu	353	0,7	0,25	84,7	1200
12	Tepung Terigu	365	8,9	1,3	77,3	1600
13	Ubi Jalar	323	1,8	0,7	27,9	800
14	Kacang Ijo	345	22,2	1,2	62,9	800
15	Kecap	46	5,7	1,3	9	1800
16	Tahu	68	7,8	4,6	1,6	2000
17	Tempe Kedelai	149	18,3	4	12,7	2000
18	Corned Beef	241	16	25	0	1770
19	Daging Sapi	207	18,8	14	0	4500
20	Sosis Sapi	452	14,5	42,3	2,3	3500
21	Telur Ayam	162	12,8	11,5	0,7	1500
22	Telur Bebek	189	13,1	14,3	0,8	2000
23	Ikan Bandeng	129	20	4,8	0	1800
24	Ikan Bawal	96	19	1,7	0	1000
25	Ikan Asin (kering)	193	42	1,5	0	3000
26	Ikan Kakap	92	20	0,7	0	2200
27	Kepiting	151	13,8	3,8	14,1	2400
28	Rebon (Segar)	81	16,2	1,2	0,7	2400
29	Sarden	338	21,1	27	1	800
30	Teri Kering	170	33,4	3	0	600
31	Udang (Segar)	295	62,4	2,3	1,8	2000
32	Bayam	36	3,5	0,5	6,5	300
33	Bayam Merah	51	4,6	0,5	1	500
34	Bawang Bombay	45	1,4	0,2	10,3	300
35	Bawang Merah	39	1,5	0,3	0,2	1000
36	Bawang Putih	95	4,5	0,2	23,1	700
37	Buncis	35	2,4	0,2	7,7	300
38	Daun Bawang	29	1,8	0,7	5,2	1000
39	Daun Pepaya	79	8	2	11,9	1000
40	Daun Singkong	73	6,8	1,2	13	1000
41	Jagung muda	33	2,2	0,1	7,4	1500
42	Jamur Kuping (Segar)	15	3,8	0,6	0,9	1180
43	Kangkung	29	3	0,3	5,4	1000

44	Ketimun	12	0,7	0,1	2,7	1000
45	Kembang Kol	25	2,4	0,2	4,9	500
46	Labu Siam	26	0,6	0,1	6,7	1000
47	Lobak	19	0,9	0,1	4,2	800
48	Pare	29	1,1	0,3	6,6	500
49	Peterseli	50	3,7	1	9	500
50	Rebung	27	2,6	0,3	5,2	500
51	Selada	15	1,2	0,2	2,9	400
52	Selada Air	17	1,7	0,3	3	400
53	Minyak Kelapa	870	1	98	0	2900
54	Minyak Kelapa Sawit	902	0	100	0	2200
55	Roti Putih	248	8	1,2	50	1400
56	Biji Jambu Monyet	562	21,6	49,6	23,6	4400
57	Kacang Merah	336	23,1	1,7	59,5	1000
58	Kacang Tanah	559	26,9	442	23,6	1000
59	Santan	324	4,2	34,3	5,6	3000
60	Babat	113	17,6	4,2	0	3000
61	Ikan Gabus	74	25,2	1,7	0	5400
62	Kerang	59	8	1,1	3,6	1000
63	Jantung Pisang	31	1,2	0,3	7,1	800
64	Kacang Kapri	98	6,7	0,4	17,7	500
65	Apel	58	0,3	0,4	14,9	1000
66	Telur Puyuh	168	12,3	12,7	1,2	1000
67	Manggis	63	0,6	0,6	15,6	1500
68	Belimbing	36	0,4	0,4	8,8	2000
69	Semangka	28	0,5	0,2	6,9	900
70	Cumi-cumi	75	16,1	0,7	0,1	1500
71	Ikan Mujair	89	18,7	1	0	3000
72	Ceker Ayam	245	20,8	17,1	2,1	3500
73	Pepaya	46	0,5	0	12,2	1000
74	Nanas	52	0,4	0,2	13,7	1500
75	Salak	77	0,4	0	20,9	1000
76	Jeruk Bali	48	0,6	0,2	12,4	2290
77	Sirsak	65	1	0,3	16,3	1900
78	Jeruk Manis	45	0,9	0,2	11,2	1700
79	Anggur	50	0,5	0,2	12,8	2300
80	Keju	326	22,8	20,3	13,1	2400
81	Abon Sapi	212	18	10,6	59,3	3400
82	Kedelai	381	40,4	16,7	24,9	1000
83	Beras Merah	359	7,5	0,9	77,6	1450
84	Beras Ketan Hitam	356	7	0,7	78	1000
85	Maizena	343	0,3	0	85	3500
86	Tepung Jagung	355	9,2	3,9	73,3	3500
87	Ampas Tahu	414	26,6	18,3	41,3	1000
88	Jengkol	20	3,5	0,1	3,1	2000
89	Daging Domba	206	17,1	14,8	0	3500
90	Hati Sapi	136	19,7	3,2	6	4000
91	Otak	125	10,4	8,6	0,8	4500
92	Pindang Layang	153	30	2,8	0	2500

93	Daun Labu Siam	60	4	0,4	4,7	1000
94	Daun Labu Waluh	30	3,6	0,6	4,5	1200
95	Labu Air	17	0,6	0,2	3,8	1200
96	Margarin	720	0,6	81	0,4	1400
97	Minyak ikan	902	0	100	0	1400
98	Minyak Kelapa	870	1	98	0	1200
99	Singkong	146	1,2	0,3	34,7	800
100	Talas	98	1,9	0,2	23,7	900
101	Emping	345	12	1,5	71,5	1800
102	Kacang Panjang	357	17,3	1,5	70	200
103	Kacang Tolo	342	22,9	1,4	61,6	600
104	Putih Telur Ayam	50	10,8	0	0,8	1000
105	Telur Asin (Bebek)	195	13,6	13,6	1,4	2500
106	Kerupuk Ikan	342	16	0,4	65,6	500
107	Kerupuk Udang	359	17,2	0,6	68,2	500
108	Belut	303	14	27	0	2000
109	Teri (Segar)	77	16	1	0	1000
110	Bengkoang	55	1,4	0,2	12,8	400
111	Daun Melinjo	99	5	1,3	21,3	400
112	Daun Oyong	22	1	0,1	5,3	300
113	Daun Pakis	35	4	0,3	6,4	400
114	Daun Petai Cina	128	12	6,5	12,4	200
115	Kembang Turi	44	1,8	0,6	9,6	200
116	Labu Waluh	29	1,1	0,3	6,6	1200
117	Melinjo	66	5	0,7	13,3	1000
118	Nangka Muda	51	2	0,4	11,3	400
119	Sawi	22	2,3	0,3	4	200
120	Minyak Wijen	902	0	100	0	1200
121	Sari Kedelai Bubuk	344	30	20	43	1500
122	Kucai	45	2,2	0,3	10,3	800
123	Daun Kecipir	47	5	0,5	8,5	500
124	Pisang raja	120	1,2	0,2	31,8	2500
125	Pisang mas	127	1,4	0,2	33,6	1500