

**OPTIMASI MENU MAKANAN ATLET
BERDASARKAN JADWAL LATIHAN
MENGUNAKAN ALGORITME GENETIKA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
M. Dimas Setiawan S
NIM:145150201111117



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGESAHAN

OPTIMASI MENU MAKANAN ATLET
BERDASARKAN JADWAL LATIHAN
MENGUNAKAN ALGORITME GENETIKA

SKRIPSI


Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

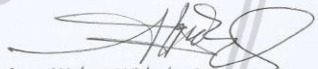
Disusun Oleh :
M. Dimas Setiawan S
NIM: 14515020111117

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Mei 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


Budi Darma Setiawan, S.Kom, M.Cs
NIP. 19841015 201404 1 002


Agus Wahyu Widodo, S.T, M.Cs
NIP. 19740805 200112 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



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

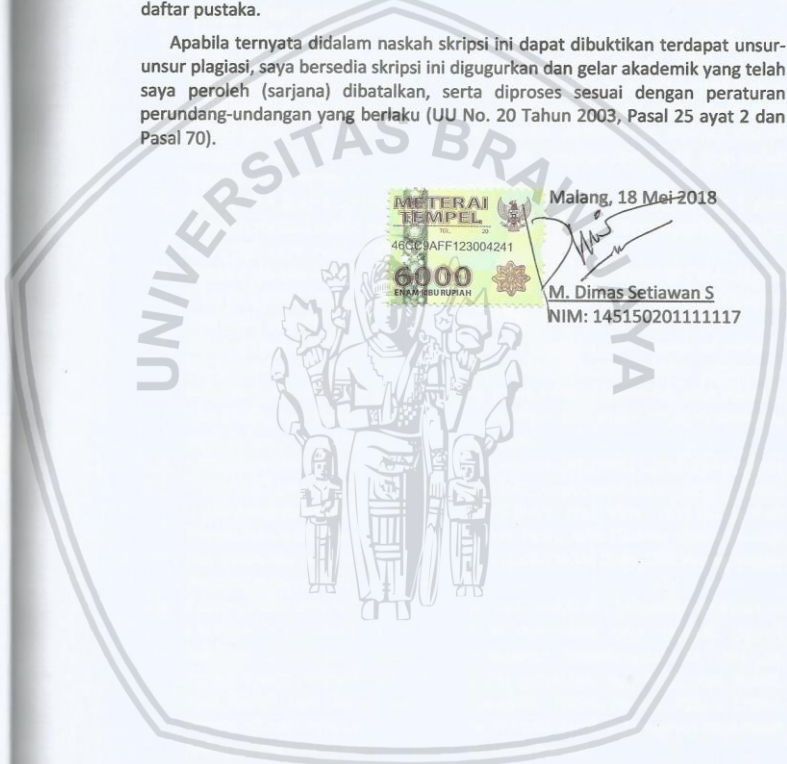


PERNYATAAN ORISINALITAS

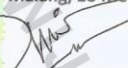
PERNYATAAN ORISINALITAS

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

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiaris, 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, 18 Mei-2018



M. Dimas Setiawan S
NIM: 145150201111117



KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena atas segala rahmat dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “*Optimasi Menu Makanan Atlet Berdasarkan Jadwal Latihan Menggunakan Algoritme Genetika*”.

Dalam penyusunan skripsi ini juga tak lepas dari bantuan semua pihak baik berupa bimbingan, kritik, saran, motivasi maupun doa. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Sanapiah dan Ibu Mastambuan selaku kedua orang tua penulis yang telah berupaya menyekolahkan anaknya dengan penuh kasih sayang, dukungan, motivasi pengorbanan materil maupun non materil. Rasa terima kasih yang tiada tara untuk kedua orang tua penulis. Tidak lupa juga kepada abang-abang dan adik penulis yang selalu mendukung penulis dalam menyelesaikan skripsi ini ketika dalam rasa tertekan dan mulai lupa untuk mengerjakan skripsi ini.
2. Bapak Budi Darma Setiawan, S.Kom, M.Cs selaku dosen pembimbing 1 atas segala ilmu dan waktu yang diluangkan dalam memberikan bimbingan serta kritik dan saran yang diberikan kepada penulis.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku dosen pembimbing 2 atas segala ilmu dan waktu yang diluangkan dalam memberikan bimbingan serta kritik dan saran yang diberikan kepada penulis.
4. Keluarga besar Kelompok Riset Mahasiswa Fakultas Ilmu Komputer periode 2015/2016 yang telah menerima saya menjadi bagian dalam perjalanan kelembagaan ini sehingga merasakan susah senang bersama, mas Hendra, mas Galih, mas Galuh, mas Haydar, mas Nizar, dan mas-mas dan mba-mba yang belum disebutkan terima kasih banyak.
5. Keluarga besar Kelompok Riset Mahasiswa Fakultas Ilmu Komputer periode 2016/2017 yang telah mengajak saya dalam pengurusan sehingga mendapat bekal pelajaran dikehidupan nanti, mas Jihad, mas Rizqy, mas Fadly, mas Yayak, mas Adit, mas Adi, mba Upik, mba Putri, mba Chusna, dan mas-mas dan mba-mba yang bagian dari pengurusan K-RISMA 2016/2017 yang belum disebutkan terima kasih banyak.
6. Keluarga departemen Administrasi dan seluruh Pengurus Himpunan Masiswa Informatika 2015 yang telah menerima saya menjadi bagian keluarga Cipul untuk membantu mengurus kelembagaan ini sehingga mengerti betapa pentingnya berbagi dan berkorban, mas Echa, mas Samid, mba Afifah, mba Laisa, Eka, Yane, Anggita, Yudha, Maria, Indah, Lisma, Nanda, dan mas-mas dan mba-mba serta teman-teman pengurus himpunan lainnya yang belum disebutkan terima kasih banyak.
7. Keluarga divisi bridge BIOS yang telah mengajar dan mengajak saya dalam berpartisipasi untuk membela Fakultas Ilmu Komputer diajang Olimpiade Brawijaya tahun 2015. Khususnya kepada Revinda Bertananda ketua divisi Bridge BIOS 2017 yang selalu menyemangati saya dalam mengerjakan skripsi ini agar tidak lupa untuk dikerjakan sehingga tidak lewat dari timeline,

terima kasih banyak. Teman-teman bridge lainnya juga mas Helto, mas Agung, mas Amir, Ibrahim partnerku, Naufal, Dewi, dan lainnya yang telah bersama-sama dalam meluangkan waktu untuk belajar dan berkembang bersama terima kasih banyak.

8. Keluarga besar Kelompok Riset Mahasiswa Fakultas Ilmu Komputer periode 2017/2018 yang telah mempercayakan kepada saya sebagai ketua dalam pengurusan periode ini, banyak hal yang saya dapatkan bersama kalian yang pastinya tidak akan terlupakan dan menjadi pelajaran yang sangat berarti untuk saya kedepannya. Dedin wakilku, Bintang dan Yani sekretaris, Nelli dan Regita bendahara, Reza dan Melli kadiv wakadiv PSDM, Wahyu dan Fiqah kadiv wakadiv Humas, Luthfi, Hanif, dan Arka kadiv kabid Litbang, dan pengurus lainnya serta anggota tercinta yang tidak dapat saya sebutkan satu persatu disini. Terima kasih yang sebanyak-banyaknya atas perjuangan kita bersama.
9. Teman-teman seperjuangan dalam mengerjakan skripsi dengan dosen pembimbing yang sama Indah Larasati, Talitha Raissa, Kholi, Ni'mah, Rosi, dan teman-teman lainnya.
10. Teman-teman kos mertojoyo yang turut menyemangati dalam pengerjaan skripsi ini, Akbar, Tri, Fajar, Hadi, Aldzha, Akbar bawah, Sany, Fikri, Yudha, dan Adit. Alhamdulillah karena semangat dari kalian saya dapat menyelesaikan skripsi ini.
11. Seluruh bapak ibu dosen dan staff Fakultas Ilmu Komputer atas segala bantuan dan ilmu yang diajarkan kepada penulis.
12. Rekan-rekan Fakultas Ilmu Komputer atas segala bantuan yang diberikan kepada penulis selama berada di Fakultas Ilmu Komputer.

Penulis menyadari skripsi ini masih jauh dari sempurna dan masih banyak terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 18 Mei 2018

Penulis
dimsetiawan_toke@yahoo.com

ABSTRAK

Muhammad Dimas Setiawan Sanapiah. 2018. Optimasi Menu Makanan Atlet Berdasarkan Jadwal Latihan Menggunakan Algoritme Genetika.

Dosen Pembimbing: Budi Darma Setiawan, S.Kom, M.Cs dan Agus Wahyu Widodo, S.T, M.Cs

Seorang atlet untuk mencapai prestasi yang maksimal pada suatu cabang olahraga yang digeluti, memerlukan sistem pelatihan yang optimal, termasuk ketersediaan dan kecukupan gizi yang sesuai dengan jenis olahraganya. Setiap jenis olahraga dengan jenis latihan yang berbeda-beda akan menghasilkan kebutuhan energi yang berbeda pula. Pemenuhan asupan gizi merupakan kebutuhan dasar bagi atlet. Berdasarkan teori olahraga dijelaskan bahwa gizi dan latihan fisik menghasilkan prestasi. Namun demikian sebagian besar asupan gizi atlet tidak tepat karena kurangnya pengetahuan dan pemahaman atlet dalam memilih makanan dan kurangnya edukasi tentang pentingnya gizi olahraga prestasi bagi atlet. Seharusnya, dengan seiring perkembangan Ilmu Pengetahuan dan Teknologi pengolahan dan manajemen kebutuhan gizi atlet dapat dilakukan dengan lebih baik. Salah satu bentuk pengolahan dan manajemen menggunakan teknologi yaitu membuat rekomendasi menu makanan untuk atlet yang optimal. Pembuatan rekomendasi menu makanan yang optimal ini dapat menggunakan algoritme genetika. Algoritme genetika merupakan algoritme yang berbasis populasi yang dapat menyelesaikan permasalahan yang berkaitan dengan optimasi dengan ruang pencarian yang sangat luas. Algoritme genetika menggunakan konsep *natural selection* dalam melakukan proses pencarian dan optimasi. Algoritme genetika banyak dimanfaatkan dalam melakukan optimasi penyusunan menu makanan. Dalam penyusunan kromosom digunakan representasi bilangan integer, metode *crossover* yang digunakan adalah *one cut point crossover*, metode mutasi yang digunakan adalah *random mutation* dan proses seleksi menggunakan metode seleksi *elitism*. Hasil yang direkomendasikan adalah menu makanan selama lima hari. Pada hasil pengujian dalam permasalahan ini didapatkan ukuran generasi optimal sebesar 450, ukuran populasi optimal sebesar 70 dan kombinasi nilai *cr* dan *mr* optimal adalah 0,5 dan 0,5. Berdasarkan hasil tersebut sistem mampu memberikan rekomendasi makanan yang memenuhi kebutuhan gizi harian.

Kata kunci: algoritme genetika, atlet prestasi, menu makanan, optimasi.

ABSTRACT

Muhammad Dimas Setiawan Sanapiah. 2018. Optimization of Athlete Food Menu Based on Exercise Schedule Using Genetic Algorithm.

Advisor: Budi DarmaSetiawan, S. Kom, M.Cs and Agus Wahyu Widodo, S.T, M.Cs

To achieve a maximum achievement in sports, an athlete needs an optimal training system, including the availability and adequacy of nutrition that corresponds with the type of exercise. Fulfilling nutrition intake is a basic need for athletes. Based on sports theory, it is explained that nutrition and physical exercise can lead to achievement. However, most of an athlete's nutrition intake is inappropriate because of the lack of an athlete's knowledge in choosing food and lack of knowledge about the importance of sports nutrition achievement. With the development of science and technology management, an athlete's nutrition intake should be done better. One form of processing and management using technology is to make recommendations of optimal food menus for athletes. This optimal food menu recommendation can use a genetic algorithm. A genetic algorithm is an algorithm based on a population that can solve a problem related to optimization with a large search space. A genetic algorithm uses a natural selection concept in the search and optimization process. A genetic algorithm is often used in optimizing the preparation of a food menu. In the preparation of chromosomes, the representation of integer numbers is used, the crossover method used is one-point crossover, the mutation method used is a random method, and the selection process uses an elitism selection method. The recommendation result is a food menu for five days. In the test results for this problem, an optimal generation size of 450, an optimal population size of 70, and the combination of cr and mr optimal values of 0.5 and 0.5 were obtained. Based on these results, the system is able to provide food recommendations that meet daily nutritional needs.

Keywords: Genetic Algorithm, Athlete Performance, Food Menu, Optimization.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR PERSAMAAN	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian pustaka	5
2.2 Kebutuhan gizi atlet	7
2.2.1 Energi	7
2.2.2 Kebutuhan zat gizi	10
2.3 Algoritme genetika	11
2.3.1 Struktur algoritme genetika	12
BAB 3 METODOLOGI	18
3.1 Tahap penelitian	18
3.1.1 Studi literatur	19
3.1.2 Pengumpulan data	19
3.1.3 Analisis dan perancangan sistem	19

3.1.4 Implementasi sistem	19
3.1.5 Pengujian sistem	20
3.1.6 Penarikan kesimpulan	20
BAB 4 PERANCANGAN	21
4.1 Formulasi permasalahan	21
4.2 Tahapan algoritme genetika	21
4.2.1 Inisialisasi populasi awal	22
4.2.2 <i>Crossover</i>	24
4.2.3 Mutasi	25
4.2.4 Evaluasi.....	26
4.2.5 Seleksi.....	28
4.3 Contoh perhitungan manual.....	29
4.3.1 Contoh permasalahan.....	29
4.3.2 Inisialisasi parameter algoritme genetika.....	31
4.3.3 Representasi kromsوم.....	31
4.3.4 Inisialisasi populasi awal	35
4.3.5 <i>Crossover</i>	36
4.3.6 Mutasi	37
4.3.7 Evaluasi.....	38
4.3.8 Seleksi.....	42
4.4 Perancangan antarmuka	42
4.4.1 Perancangan antarmuka halaman <i>input</i> data.....	42
4.4.2 Perancangan antarmuka halaman algoritme genetika.....	43
4.4.3 Perancangan antarmuka halaman hasil.....	44
4.4.4 Perancangan halaman lampiran	45
4.5 Perancangan pengujian.....	46
4.5.1 Pengujian kombinasi <i>crossover rate</i> dan <i>mutation rate</i>	47
4.5.2 Pengujian ukuran populasi.....	47
4.5.3 Pengujian ukuran generasi.....	48
BAB 5 IMPLEMENTASI	49

5.1 Spesifikasi sistem.....	49
5.1.1 Spesifikasi perangkat keras	49
5.1.2 Spesifikasi perangkat lunak.....	49
5.2 Implementasi program.....	49
5.2.1 Implementasi proses inialisasi populasi awal.....	50
5.2.2 Implementasi proses <i>crossover</i>	51
5.2.3 Implementasi proses mutasi	52
5.2.4 Implementasi proses perhitungan <i>fitness</i>	53
5.2.5 Implementasi proses seleksi	54
5.3 Implementasi antarmuka	55
5.3.1 Implementasi antarmuka halaman <i>input data</i>	55
5.3.2 Implementasi antarmuka halaman algoritme genetika	56
5.3.3 Implementasi antarmuka halaman hasil.....	57
5.3.4 Implementasi antarmuka halaman lampiran.....	58
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	60
6.1 Pengujian kombinasi <i>crossover rate</i> dan <i>mutation rate</i>	60
6.2 Pengujian ukuran populasi.....	61
6.3 Pengujian ukuyran generasi.....	63
6.4 Hasil analisis global	64
BAB 7 PENUTUP	67
7.1 Kesimpulan.....	67
7.2 Saran.....	67
DAFTAR PUSTAKA.....	69
LAMPIRAN A DATA ATLET	70
LAMPIRAN B DATA MAKANAN.....	72
LAMPIRAN C REKOMENDASI MENU MAKANAN	77



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	6
Tabel 2.2 Kategori Tingkat Aktivitas Fisik.....	9
Tabel 2.3 Kebutuhan Energi (kalori/menit) pada Berbagai Jenis Olahraga	9
Tabel 4.1 Sumber Karbohidrat	21
Tabel 4.2 Jadwal Latihan Atletik.....	30
Tabel 4.3 Representasi Kromosom	32
Tabel 4.4 Konversi Kromosom Menjadi Menu Makanan	33
Tabel 4.5 Inisialisasi Populasi Awal	35
Tabel 4.6 Inisialisasi Populasi Awal (Lanjutan).....	36
Tabel 4.7 Proses <i>Crossover</i>	36
Tabel 4.8 <i>Offspring</i> Hasil <i>Crossover</i>	37
Tabel 4.9 Proses Mutasi	37
Tabel 4.10 <i>Offspring</i> Hasil Mutasi	38
Tabel 4.11 Populasi Baru.....	38
Tabel 4.12 Populasi Baru (Lanjutan)	39
Tabel 4.13 Jumlah Kandungan Gizi Makanan Sistem.....	39
Tabel 4.14 Jumlah Kandungan Gizi Makanan Sistem (Lanjutan)	40
Tabel 4.15 Penalti Individu.....	40
Tabel 4.16 Normalisasi Individu	41
Tabel 4.17 Nilai <i>Fitness</i> Individu.....	41
Tabel 4.18 Hasil Seleksi	42
Tabel 4.19 Hasil Perhitungan <i>Fitness</i> Generasi Kedua.....	42
Tabel 4.20 Pengujian <i>Crossover Rate</i> dan <i>Mutation Rate</i>	47
Tabel 4.21 Pengujian Ukuran Populasi	48
Tabel 4.22 Pengujian Ukuran Generasi.....	48
Tabel 5.1 Spesifikasi Perangkat Keras	49
Tabel 5.2 Spesifikasi Perangkat Lunak	49
Tabel 6.1 Hasil Pengujian Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	60
Tabel 6.2 Hasil Pengujian Ukuran Populasi.....	61
Tabel 6.3 Hasil Pengujian Ukuran Generasi	63
Tabel 6.4 Kandungan Gizi Sistem	65
Tabel 6.5 Perbandingan Kebutuhan Gizi Aktual dan Sistem.....	65

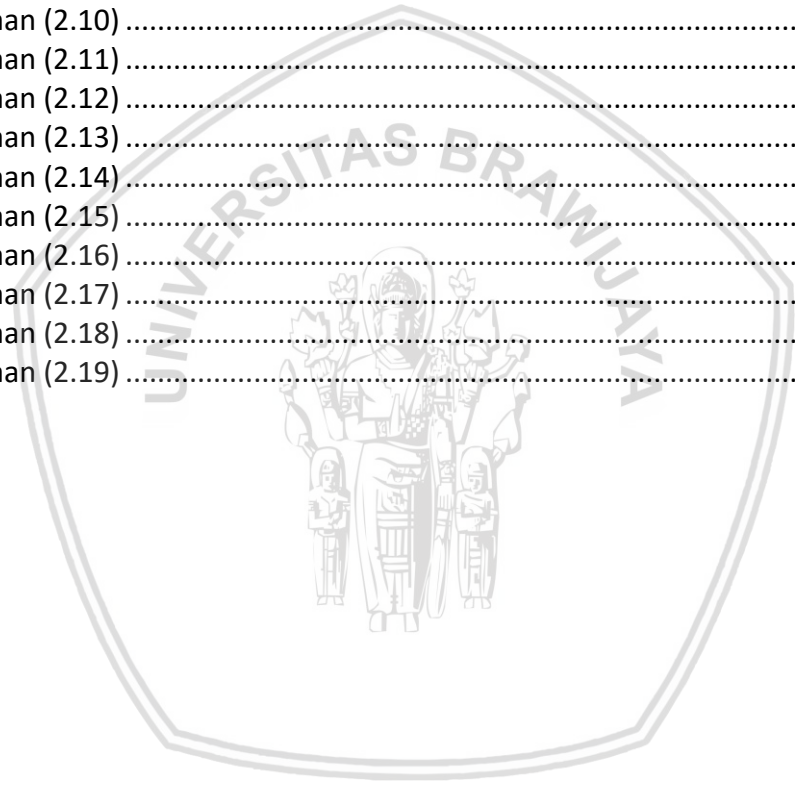
DAFTAR GAMBAR

Gambar 2.1 Pencarian Solusi dengan Algoritme Genetika	12
Gambar 3.1 Blog Diagram Metodologi Penelitian	18
Gambar 4.1 Diagram Alir Algoritme Genetika	21
Gambar 4.2 Diagram Alir Algoritme Genetika (Lanjut).....	22
Gambar 4.3 Diagram Alir Inisialisasi Populasi Awal	23
Gambar 4.4 Diagram Alir Proses <i>Crossover</i>	24
Gambar 4.5 Diagram Alir Proses Mutasi.....	25
Gambar 4.6 Diagram Alir Proses Mutasi (Lanjut)	26
Gambar 4.7 Diagram Alir Evaluasi.....	27
Gambar 4.8 Diagram Alir Seleksi Elitism	28
Gambar 4.9 Diagram Alir Seleksi Elitism (Lanjut).....	29
Gambar 4.10 <i>Input</i> Data Atlet.....	43
Gambar 4.11 <i>Input</i> Data Algoritme Genetika	43
Gambar 4.12 Halaman Algoritme Genetika Inisialisasi.....	44
Gambar 4.13 Halaman Algoritme Genetika Reproduksi.....	44
Gambar 4.14 Halaman Hasil Individu Terbaik.....	45
Gambar 4.15 Halaman Hasil Rekomendasi Menu Makanan	45
Gambar 4.16 Halaman Lampiran Data Makanan.....	46
Gambar 4.17 Halaman Lampiran Data Latihan.....	46
Gambar 5.1 Implementasi Kode Program Inisialisasi Populasi Awal.....	50
Gambar 5.2 Implementasi Kode Program Proses <i>Crossover</i>	51
Gambar 5.3 Implementasi Kode Program Proses Mutasi.....	52
Gambar 5.4 Implementasi Kode Program Proses Perhitungan <i>Fitness</i>	54
Gambar 5.5 Implementasi Kode Program Proses Seleksi	54
Gambar 5.6 Antarmuka Halaman <i>Input</i> Atlet	55
Gambar 5.7 Antarmuka Halaman <i>Input</i> Algoritme Genetika	56
Gambar 5.8 Antarmuka Halaman Algoritme Genetika Inisialisasi.....	56
Gambar 5.9 Antarmuka Halaman Algoritme Genetika Reproduksi.....	57
Gambar 5.10 Antarmuka Halaman Hasil Individu Terbaik.....	57
Gambar 5.11 Antarmuka Halaman Hasil Rekomendasi Menu Makanan	58
Gambar 5.12 Antarmuka Halaman Lampiran Data Makanan.....	58
Gambar 5.13 Antarmuka Halaman Lampiran Data Latihan.....	59
Gambar 6.1 Grafik Hasil Pengujian Kombinasi <i>Crossover Rate</i> dan <i>Mutatiojn Rate</i> ..	61
Gambar 6.2 Grafik Hasil Pengujian Ukuran Populasi	62
Gambar 6.3 Grafik Hasil Pengujian Ukuran Generasi	64



DAFTAR PERSAMAAN

Persamaan (2.1)	7
Persamaan (2.2)	7
Persamaan (2.3)	8
Persamaan (2.4)	8
Persamaan (2.5)	14
Persamaan (2.6)	14
Persamaan (2.7)	14
Persamaan (2.8)	15
Persamaan (2.9)	15
Persamaan (2.10)	15
Persamaan (2.11)	15
Persamaan (2.12)	15
Persamaan (2.13)	15
Persamaan (2.14)	15
Persamaan (2.15)	15
Persamaan (2.16)	15
Persamaan (2.17)	16
Persamaan (2.18)	16
Persamaan (2.19)	16



DAFTAR LAMPIRAN

LAMPIRAN A DATA ATLET	70
LAMPIRAN B	72
B.1 Sumber Karbohidrat	72
B.2 Sumber Hewani	73
B.3 Sumber Nabati.....	74
B.4 Sayuran	75
B.5 Pelengkap	76
LAMPIRAN C REKOMENDASI MENU MAKANAN	77





BAB 1 PENDAHULUAN

1.1 Latar belakang

Perkembangan olahraga Indonesia sudah cukup berkembang mengikuti perkembangan negara maju Asia, Eropa, maupun benua paman sam Amerika. Seperti yang dapat kita lihat Indonesia selalu ikut dalam beberapa cabang lomba yang diperlombakan pada kompetisi olahraga tertinggi dunia yaitu olimpiade. Indonesia sendiri dalam perhelatan olimpiade sejak tahun 1988 di Seoul Korea Selatan sampai tahun 2016 kemarin di Rio de Janeiro Brazil selalu membawa pulang medali. Keberhasilan Indonesia dalam membawa pulang medali ini tidak lepas dari perjuangan atlet selama bertanding.

Seorang atlet untuk mencapai prestasi yang maksimal pada suatu cabang olahraga yang digeluti, memerlukan sistem pelatihan yang optimal, termasuk ketersediaan dan kecukupan gizi yang sesuai dengan jenis olahraganya. Setiap jenis olahraga dengan jenis latihan yang berbeda-beda akan menghasilkan kebutuhan energi yang berbeda pula. Untuk meningkatkan prestasi atlet Indonesia ke depan, dirasakan perlu untuk memperbaiki dan menyempurnakan sistem pembinaan dan pelatihan olahraga, terutama dalam melakukan pendekatan dan penerapan Ilmu Pengetahuan dan Teknologi (IPTEK), olahraga termasuk gizi olahraga (Kemenkes, 2014).

Pemenuhan asupan gizi merupakan kebutuhan dasar bagi atlet. Berdasarkan teori olahraga dijelaskan bahwa gizi dan latihan fisik menghasilkan prestasi. Bahkan federasi sepak bola dunia telah mengeluarkan pernyataan bahwasanya gizi berperan dalam keberhasilan satu tim. Namun demikian sebagian besar asupan gizi atlet tidak tepat karena kurangnya pengetahuan dan pemahaman atlet dalam memilih makanan, kurangnya edukasi tentang pentingnya gizi olahraga prestasi bagi atlet. Pelatih, pengurus serta kurangnya ketersediaan tenaga gizi dan kesehatan yang memahami dan memiliki kompetensi dalam ilmu gizi olahraga prestasi. Peranan gizi dalam olahraga prestasi menuntut tenaga gizi dan kesehatan yang terampil untuk menjaga secara khusus dan intensif kebutuhan zat gizi atlet (Kemenkes, 2014).

Namun tidak seperti harapan yang dikemukakan di atas, beberapa permasalahan yang pernah terjadi dikarenakan kesalahan manajemen dalam pengolahan sarana dan prasarana atlet seperti yang diberitakan oleh GoRiau.com yang berjudul "Atlet Panjat Tebing asal Meranti Gadai Hape untuk Biaya Makan di Pekanbaru" ini memberitakan tentang kondisi atlet yang tengah berjuang dalam latihannya namun tidak mendapatkan bantuan sepersenpun dari lembaga terkait. Berbeda lagi dengan kasus yang dialami oleh atlet pelatda asal Sulawesi dimana yang diberitakan oleh jppn.com yang berjudul "Makanan Kurang, Atlet Teriak, Piring pun Dibalikkan" memberitakan tentang menu yang disajikan oleh pengelola makanan tidak bergizi dan tidak cukup untuk konsumsi semua atlet dan pelatih.

Seharusnya, dengan seiring perkembangan Ilmu Pengetahuan dan Teknologi pengelolaan dan manajemen kebutuhan gizi atlet dapat dilakukan dengan lebih baik. Salah satu bentuk pengolahan dan manajemen menggunakan teknologi yaitu membuat rekomendasi menu makanan untuk atlet yang optimal. Pembuatan rekomendasi menu makanan yang optimal ini dapat menggunakan algoritme genetika. Algoritme genetika merupakan algoritme yang berbasis populasi yang dapat menyelesaikan permasalahan yang berkaitan dengan optimasi dengan ruang pencarian yang sangat luas. Algoritme genetika menggunakan konsep *natural selection* dalam melakukan proses pencarian dan optimasi. Algoritme genetika dapat menyelesaikan masalah dengan memberikan himpunan solusi dan menemukan solusi yang paling optimal. Algoritme genetika juga dapat menyelesaikan masalah yang bersifat kompleks yang membutuhkan banyak variabel (Mahmudy, 2013).

Penerapan algoritme genetika dapat dilakukan pada berbagai macam masalah yang membutuhkan solusi optimum. Penelitian sebelumnya yang dilakukan oleh Wahid et al. (2015) pada optimasi komposisi makanan untuk penderita kolesterol. Digunakan 100 bahan makanan pada penelitian tersebut dengan panjang kromosom 9 menghasilkan solusi optimal pada populasi berukuran 100 dengan rata-rata nilai *fitness* sebesar 0,1862463, ukuran generasi 90, nilai *cr* = 0,7 dan *mr* = 0,3 mendapatkan hasil rata-rata nilai *fitness* sebesar 0,18575847. Penelitian lain juga pernah dilakukan oleh Maryamah et al.(2017) pada optimasi komposisi makanan penderita diabetes dan komplikasinya dengan harga terjangkau. Hasil pada penelitian tersebut adalah komposisi makanan dalam satu hari yang terdiri dari makan pagi, makan siang, makan malam. Komposisi makanan yang dihasilkan telah sesuai dengan kebutuhan gizi serta dengan harga terjangkau.

Berdasarkan permasalahan di atas, penulis melakukan penelitian dengan judul “Optimasi Menu Makanan Atlet Berdasarkan Jadwal Latihan Menggunakan Algoritme Genetika”. Penelitian ini akan memberikan keluaran berupa menu makanan bagi atlet, sehingga dapat meningkatkan kualitas manajemen pengelolah makanan atlet dalam memberikan makanan kepada atlet semasa latihan.

1.2 Rumusan masalah

Berdasarkan latar belakang yang ada, maka rumusan masalah yang dapat dibuat adalah sebagai berikut:

1. Bagaimana pengaruh perubahan nilai parameter genetika dalam optimasi pembagian menu makanan atlet berdasarkan jadwal latihan?
2. Bagaimana kualitas solusi yang diperoleh dari penerapan algoritme genetika pada optimasi menu makanan atlet berdasarkan jadwal latihan?

1.3 Tujuan

Dari rumusan masalah di atas, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui pengaruh perubahan nilai parameter genetika dalam optimasi pembagian menu makanan atlet berdasarkan jadwal latihan.
2. Mengetahui kualitas solusi yang diperoleh dari penerapan algoritme genetika pada optimasi menu makanan atlet berdasarkan jadwal latihan.

1.4 Manfaat

Manfaat dari penelitian ini antara lain:

1. Dapat memberikan rekomendasi menu makan atlet yang sesuai dengan latihannya.
2. Dapat mengembangkan sistem yang membantu lembaga olahraga untuk mengelola keuangan dalam hal makan atlet selama *training center*.

1.5 Batasan masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini dibuat hanya untuk mengetahui menu makanan atlet berdasarkan jadwal atlet yang telah dimasukkan didatabase sistem terlebih dahulu.
2. Nilai kebutuhan energi atlet berdasarkan jadwal latihan telah dikonversi terlebih dahulu dan dimasukkan ke dalam sistem.
3. Hasil optimasi dari penerapan metode digunakan untuk menentukan bentuk yang paling optimal untuk pembuatan menu makanan yang ditujukan untuk atlet.
4. Pengujian hanya fokus pada pengujian metode, tidak pada pengujian kinerja hardware.

1.6 Sistematika pembahasan

1. BAB 1 PENDAHULUAN

Bab ini membahas latar belakang masalah, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan.

2. BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai dasar teori yang digunakan sebagai acuan dalam melakukan penelitian. Pada bab ini juga membahas penelitian-penelitian mengenai algoritme genetika yang telah dilakukan sebelumnya. Dasar teori yang digunakan sebagai pedoman dalam melakukan penelitian ini meliputi kebutuhan gizi atlet dan algoritme genetika.

3. BAB 3 METODOLOGI

Bab ini membahas deskripsi alur proses penelitian yang akan dilakukan meliputi studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi sistem, pengujian sistem, dan penarikan kesimpulan. Pada bab ini juga dijelaskan mengenai kebutuhan sistem yang diperlukan dalam penelitian meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4. BAB 4 PERANCANGAN

Bab ini membahas rancangan sistem berupa alur algoritme sistem dan rancangan pengujian sistem.

5. BAB 5 IMPLEMENTASI

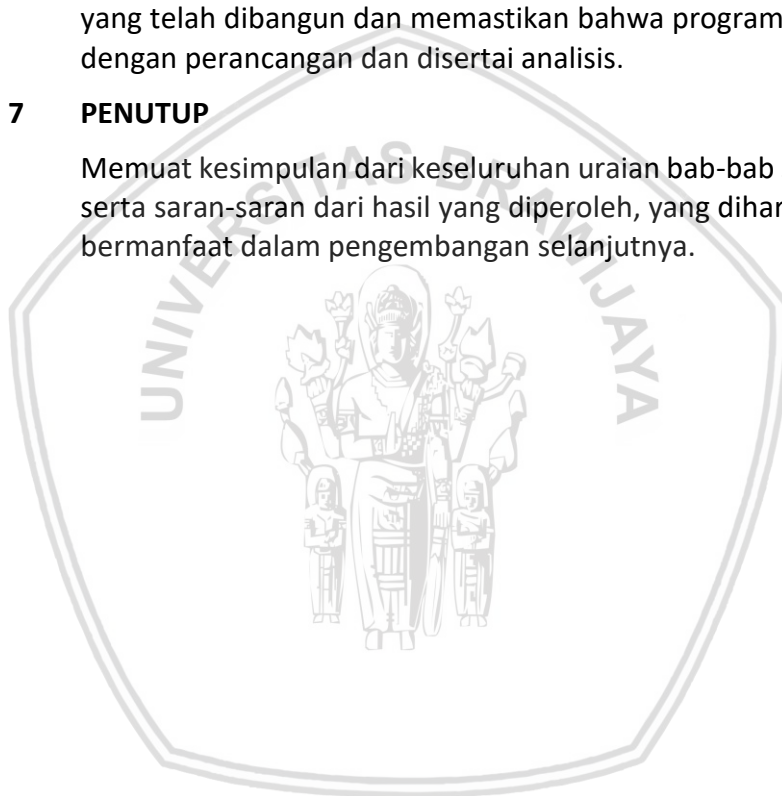
Bab ini membahas proses implementasi algoritme genetika untuk optimasi menu makanan atlet yang sesuai dengan perancangan yang telah dibuat sebelumnya.

6. BAB 6 PENGUJIAN DAN PEMBAHASAN

Bab ini membahas proses dan hasil dari pengujian terhadap sistem yang telah dibangun dan memastikan bahwa program telah sesuai dengan perancangan dan disertai analisis.

7. BAB 7 PENUTUP

Memuat kesimpulan dari keseluruhan uraian bab-bab sebelumnya, serta saran-saran dari hasil yang diperoleh, yang diharapkan dapat bermanfaat dalam pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini dijelaskan mengenai beberapa teori terkait kebutuhan gizi atlet dan algoritme genetika. Pada bagian kajian pustaka juga akan menjelaskan mengenai penelitian-penelitian sebelumnya menyangkut implementasi dari Algoritme Genetika.

2.1 Kajian pustaka

Kajian pustaka pada penelitian ini akan menjelaskan penelitian-penelitian yang sudah pernah dilakukan sebelumnya. Pada penelitian pertama dilakukan oleh Fitria Dwi Kusumaningsih pada tahun 2016. Pada penelitian ini melakukan optimasi kombinasi bahan makanan untuk anggota keluarga. Hasil yang diberikan adalah rekomendasi bahan makanan selama 7 hari dengan biaya minimum. Hasil rekomendasi dengan algoritme genetika tersebut berhasil menghemat biaya rata-rata sebesar 15,49% dengan parameter algoritme genetika yang tepat adalah jumlah generasi 80, populasi 80 dan nilai kombinasi cr dan mr sebesar 0,5:0,5.

Pada penelitian yang kedua dilakukan oleh Muhammad Shafaat pada tahun 2017. Penelitian tersebut melakukan optimasi pada komposisi makanan diet bagi penderita *hipertensi*. Hasil yang diberikan adalah rekomendasi makanan untuk beberapa hari yang terdiri dari makan pagi, makan siang, dan makan malam dengan biaya minimal. Pada penelitian tersebut hasil optimal diperoleh pada ukuran populasi 80 dengan nilai *fitness* 1100,495433, jumlah generasi 300 dengan nilai *fitness* sebesar 1190,022286 serta nilai $cr = 0,1$, $mr = 0,9$ dan nilai *fitness* sebesar 1150,466927.

Penelitian ketiga dilakukan oleh Maryamah pada tahun 2017. Penelitian tersebut melakukan optimasi komposisi makanan pada penderita diabetes melitus dan komplikasinya. Hasil yang diberikan adalah susunan makanan diet bergizi dengan harga terjangkau. Dalam penelitian tersebut hasil yang didapatkan adalah individu dalam populasi optimal berjumlah 250 individu, jumlah generasi 145, nilai cr dan mr paling optimal adalah 0,7 dan 0,3.

Penelitian keempat dilakukan oleh Nurabaiti Wahid pada tahun 2015. Penelitian tersebut melakukan optimasi pada komposisi makanan untuk penderita kolesterol. Hasil yang diberikan oleh penelitian tersebut membuktikan bahwa parameter algoritme genetika mempengaruhi hasil nilai *fitness*. Jumlah generasi yang semakin besar dibandingkan dengan populasi akan menyebabkan konvergensi. Penelitian tersebut mendapatkan hasil optimal pada generasi 90 dengan nilai rata-rata *fitness* 0,1838946, populasi yang optimal adalah 100 dengan nilai rata-rata *fitness* 0,1862463, kombinasi nilai cr dan mr adalah 0,7 dan 0,3.

Penelitian kelima dilakukan oleh Ayu Puspo Sari pada tahun 2014. Penelitian tersebut melakukan optimasi gizi untuk ibu hamil dengan hasil keluaran yaitu makan pagi, makan siang, dan makan malam. Namun, saat ini telah banyak ditemukan kasus ibu hamil yang menderita penyakit hipertensi, sehingga

penelitian tersebut kurang cocok digunakan dalam menyusun makanan untuk ibu hamil dengan penyakit hipertensi. Oleh karena itu sangat penting untuk melakukan penelitian optimasi asupan makanan harian bagi ibu hamil penderita hipertensi.

Berdasarkan penelitian yang sudah ada sebelumnya maka penulis mengusulkan sebuah penelitian dengan menggunakan algoritme genetika dalam optimasi menu makanan atlet berdasarkan jadwal latihan. *Input* yang akan digunakan antara lain usia atlet, berat badan, tinggi badan, aktivitas dan jadwal latihan atlet. Sistem akan menghitung besarnya energi yang dibutuhkan oleh atlet dan memberikan hasil keluaran asupan makanan harian yang sesuai selama tujuh lima. Kajian pustaka yang berkaitan dengan penelitian dijelaskan pada Tabel 2.1.

Tabel 0.1 Kajian Pustaka

No.	Pustaka	Objek	Metode	Output
1	Fitria Dwi Kusumaningsih, Imam Cholissodin, Budi Darma Setiawan, 2016	Bahan makanan	Algoritme genetika	Rekomendasi kombinasi bahan makanan untuk anggota keluarga selama tujuh hari berdasarkan harga minimum.
2	Muhammad Shafaat, Imam Cholissodin, Edy Santoso, 2017	Komposisi Makanan Penderita <i>Hipertensi</i>	Algoritme Genetika	Rekomendasi komposisi makanan untuk beberapa hari
3	Maryamah, Rekyan Regarsari Mardi Putri, Satrio Agung Wicaksono, 2017	Komposisi makanan penderita diabetes mellitus dan komplikasinya	Algoritme genetika	Komposisi makanan yang memenuhi gizi dengan harga terjangkau untuk penderita diabetes dan komplikasinya.
4	Nurbaiti Wahid, Wayan Firdaus Mahmudy, 20015	Komposisi makanan penderita kolesterol	Algoritme genetika	Rekomendasi proporsi luas lahan yang optimal untuk digunakan sebagai lahan pertanian.
5	Ayu Puspo Sari, Wayan Firdaus Mahmudy, Candra Dewi, 2014	Gizi ibu hamil	Algoritme genetika	Rekomendasi makanan dalam satu hari
6	Penulis	Menu makanan atlet	Algoritme genetika	Rekomendasi susunan makanan untuk atlet dalam lima hari terdiri dari makan pagi, makan siang, makan malam.

2.2 Kebutuhan gizi atlet

Perhitungan dan pemenuhan kebutuhan energi dan zat gizi bagi atlet harus mempertimbangkan jenis olahraga, tahapan pemenuhan gizi untuk periode latihan, kompetisi dan pemulihan. Selain itu perlu juga diperhatikan variasi makanan, kesukaan dan daya terima atlet agar asupannya dapat memenuhi kebutuhan atlet. Energi dihasilkan dari zat gizi makro yaitu karbohidrat, protein dan lemak. Nilai energi yang dihasilkan dari karbohidrat, protein dan lemak berbeda, nilai tersebut yaitu:

1 gram karbohidrat	menghasilkan	4 kkal
1 gram protein	menghasilkan	4 kkal
1 gram lemak	menghasilkan	9 kkal

Sumber: (Kemenkes, 2014)

Makanan seorang atlet harus mengandung semua zat gizi makro dan mikro. Secara umum menu makanan harus mengandung:

No.	Zat Gizi	Kandungan gizi (%)
1.	Karbohidrat	40 – 70
2.	Lemak	20 – 45
3.	Protein	12 – 20

Sumber: (Kemenkes, 2014)

2.2.1 Energi

Kebutuhan energi ditentukan oleh tiga komponen yaitu energi *Basal Metabolic Rate* (BMR), *Specific Dynamic Action* (SDA) dan Aktivitas Fisik. Selain itu terdapat juga pengeluaran energi untuk setiap jenis latihan. Dimana kebutuhan energi ini dapat dihitung menggunakan Persamaan (2.1) berikut:

$$\text{Kebutuhan Energi Harian} = (\text{BMR} + \text{SDA}) \times \text{Aktivitas} \quad (0.1)$$

(Kemenkes, 2014)

Keterangan:

BMR = *Basal Metabolic Rate*

SDA = *Specific Dynamic Action*

Sedangkan untuk menghitung kebutuhan total yang dibutuhkan dapat menggunakan Persamaan (2.2)

$$\text{Kebutuhan Energi Total} = \text{Kebutuhan Energi Harian} + \text{Pengeluaran Energi Latihan} \quad (0.2)$$

(Kemenkes, 2014)

1. **Basal Metabolic Rate (BMR)**

Basal Metabolic Rate (BMR) adalah jumlah energi yang dikeluarkan untuk aktivitas vital tubuh, seperti denyut jantung, bernafas, transmisi listrik pada otot dan syaraf, dan lain-lain. BMR untuk setiap orang dipengaruhi oleh umur, massa tubuh, komposisi tubuh dan jenis kelamin. BMR juga dipengaruhi oleh perubahan faktor lingkungan, seperti suhu, kelembaban, ketinggian tempat berlatih, dan keadaan emosi tertentu, seperti rasa takut, cemas dan ketegangan (Kemenkes, 2014)

Untuk menghitung nilai BMR ini dapat menggunakan persamaan yang dikembangkan oleh Harris dan Benedict, persamaan Harris dan Benedict dapat dilihat pada Persamaan (2.3) dan Persamaan (2.4) berikut:

$$H = 66,473 + 13,752W + 5,003S - 6,755A \text{ (bagi laki - laki)} \quad (0.3)$$

$$H = 65,5096 + 9,563W + 1,850S - 4,676A \text{ (bagi wanita)} \quad (0.4)$$

(Kemenkes, 2014)

Keterangan:

H = Kalori

W = Berat Badan (Kg)

S = Tinggi Badan (cm)

A = Umur (tahun)

2. **Specific Dynamic Action (SDA)**

Specific Dynamic Action (SDA) adalah energi yang dibutuhkan untuk mencerna zat gizi makro. Untuk mencerna karbohidrat dibutuhkan sebesar 6-7% dari BMR, untuk mencerna protein 20-30% dari BMR dan untuk mencerna lemak dibutuhkan sebesar 4-14% dari BMR. Untuk bahan makanan campuran yang terdiri dari karbohidrat, protein dan lemak SDAnya adalah 10% dari BMR (Kemenkes, 2014)

3. **Aktivitas Fisik**

Setiap aktivitas fisik memerlukan energi untuk bergerak. Pengeluaran energi untuk aktivitas fisik harian ditentukan oleh jenis, intensitas dan lama aktivitas fisik. Estimasi energi yang dikeluarkan oleh berbagai aktivitas fisik sangat sulit dilakukan secara teliti. Perhitungan kasar biasa menggunakan formulasi sebagai berikut:

Tabel 0.2 Kategori Tingkat Aktivitas Fisik

Kategori	Tingkatan Aktivitas Fisik	Koefisien Aktivitas Fisik	
		Pria	Wanita
Tidak aktif (<i>Sedentary</i>)	$\geq 1,0 - < 1,4$	1,00	1,00
Aktif Ringan (<i>Low Active</i>)	$\geq 1,4 - < 1,6$	1,11	1,12
Aktif (<i>Active</i>)	$\geq 1,6 - < 1,9$	1,24	1,27
Sangat Aktif (<i>Very Active</i>)	$\geq 1,9 - < 2,5$	1,48	1,45

Sumber: (Kemenkes, 2014)

4. Pengeluaran Energi Untuk Setiap Jenis Latihan

Pengeluaran energi untuk latihan fisik dan olahraga ditentukan oleh jenis olahraga, intensitas dan lamanya latihan fisik dan olahraga. Kebutuhan energi berdasarkan setiap jenis dan lama olahraga dapat dilihat pada Tabel 2.3.

Tabel 0.3 Kebutuhan Energi (kalori/menit) pada Beberapa Jenis Olahraga

Jenis Olahraga	Kebutuhan Kalori Berdasar Berat Badan (Kg)			
	50	60	70	80
	Balapan Sepeda: - 9 km/jam - 15 km/jam	3 3	4 6	4 7
Bulu Tangkis	5	6	7	7
Bola Basket	7	8	10	11
Bola Volly	2	3	4	4
Lari : - 5,5 menit/km - 5 menit/km - 4,5 menit/km	10 10 11	12 12 13	14 15 15	15 17 18
Renang : - gaya bebas - gaya punggung - gaya dada	8 9 8	10 10 10	11 12 11	12 13 13

Sumber: (Kemenkes, 2014)

2.2.2 Kebutuhan zat gizi

Kebutuhan zat gizi yang dibutuhkan terdiri dari karbohidrat, protein, dan lemak.

1. Karbohidrat

Karbohidrat merupakan sumber energi utama dan memegang peranan sangat penting untuk seorang atlet dalam melakukan olahraga. Untuk olahraga, energi berupa ATP dapat diambil dari karbohidrat yang terdapat dalam tubuh berupa glukosa dan glikogen yang disimpan dalam otot dan hati. Selama beberapa menit permulaan kerja glukosa dalam darah merupakan sumber energi utama, selanjutnya tubuh menggunakan glikogen otot dan hati. Glikogen otot dipergunakan langsung oleh otot untuk pembentukan energi, sedangkan glikogen hati mengalami perubahan menjadi glukosa yang akan masuk ke peredaran darah untuk selanjutnya dipergunakan oleh otot. Kebutuhan karbohidrat 40-70%. (Kemenkes, 2014).

2. Protein

Protein merupakan substrat penghasil energi yang bermakna selama berolahraga oleh karena hanya 12 - 20% dari total energi yang dibutuhkan. Protein terutama berperan sebagai "spare part" tubuh, yaitu sebagai zat pembangun komponen dan struktur jaringan tubuh, mengganti komponen dan struktur jaringan tubuh yang rusak seperti otot, serta berperan dalam pembentukan enzim, hormon, neurotransmitter dan antibodi.

Protein sangat diperlukan oleh atlet terutama pada atlet cabang olahraga yang membutuhkan kekuatan dan power karena protein membantu proses pembentukan serabut otot sehingga meningkatkan massa otot. Namun demikian, atlet olahraga endurans juga membutuhkan protein untuk membantu proses adaptasi akibat latihan, memperbaiki serabut otot yang rusak, dan pembentukan enzim-enzim. Kebutuhan protein untuk atlet berkisar antara 1,2 - 1,7 gr/kgBB/hari dengan maksimal 2 gr/kgBB/hari. Kebutuhan protein ini biasanya sudah dapat dipenuhi oleh atlet melalui makanan tinggi kalori (Kemenkes, 2014).

3. Lemak

Lemak adalah sumber energi utama pada aktivitas fisik dengan durasi lama dan intensitas rendah, seperti maraton. Dikenal beberapa jenis lemak yaitu lemak sederhana misalnya trigliserida; lemak kompleks yaitu kombinasi lemak sederhana dengan molekul lain seperti fosfor disebut sebagai fosfolipid. HDL (*high density lipoprotein*) dan LDL (*low density lipoprotein*) adalah jenis lemak yang berkombinasi dengan protein yang disebut sebagai lipoprotein. Bila mengandung sedikit lemak dan banyak protein disebut HDL dan bila mengandung banyak lemak dan kurang protein disebut LDL. Olahraga aerobik yang teratur dapat meningkatkan kadar HDL. Kolesterol dibutuhkan oleh tubuh untuk membangun membran sel, sintesis vitamin D, hormon adrenal,

estrogen dan hormon lain, serta diperlukan untuk pembentukan garam empedu (Kemenkes, 2014).

Lemak dalam makanan berasal dari tumbuhan dan hewan. Lemak tumbuhan (lemak nabati) biasanya berbentuk cair, contoh minyak kelapa, minyak sawit, minyak jagung. Lemak hewani sering dijumpai dalam bentuk padat, contoh mentega dan keju. Lemak adalah bahan makanan yang paling lama dicerna di lambung sehingga akan memperlambat rasa lapar.

Kebutuhan lemak berkisar antara 20 - 45% dari kebutuhan kalori total. Bila mengonsumsi lemak kurang 20% kurang dari kebutuhan kalori total tidak akan memberi keuntungan pada kinerja fisik. Demikian pula bila mengonsumsi lemak lebih 45% dari kebutuhan kalori total maka akan berbahaya bagi kesehatan atlet. Meskipun tidak secara langsung berperan dalam peningkatan prestasi, lemak dalam jumlah tertentu masih sangat dibutuhkan oleh tubuh untuk fungsi organ dan pembentukan hormon. Kebutuhan lemak pada atlet dianjurkan 20-45% dari total kalori yang dibutuhkan. Kebutuhan lemak ini harus dipenuhi untuk membentuk jaringan lemak. Jaringan lemak harus cukup terutama pada atlet wanita. Menstruasi dapat terjadi bila kadar lemak tubuh minimal 8%. Bila kadar lemak tubuh kurang dari 8%, maka menstruasi tidak terjadi karena rendahnya hormon estrogen. Rendahnya kadar hormon estrogen juga dapat menyebabkan osteoporosis (Kemenkes, 2014).

2.3 Algoritme genetika

Algoritme genetika (GA) merupakan salah satu cabang dari Algoritme Evolusi (EA) yang sering diterapkan ketika menemukan permasalahan yang kompleks. Algoritme genetika meniru konsep evolusi alami dari sebuah populasi individu. Algoritme genetika banyak digunakan untuk permasalahan yang berkaitan dengan ilmu fisika, biologi, ekonomi, dan ilmu lainnya yang membutuhkan metode pengoptimalan dengan model matematika yang kompleks (Mahmudy, 2013).

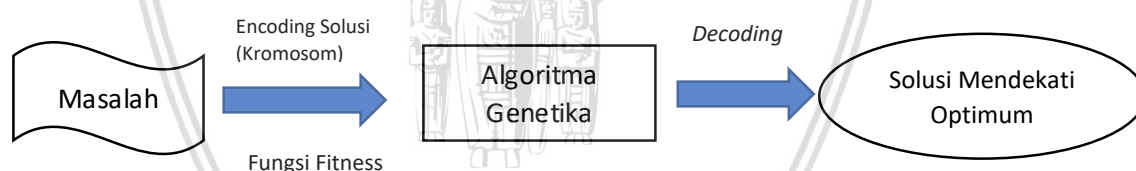
Pada algoritme genetika, proses awal yang dilakukan adalah proses inialisasi. Proses inialisasi bertujuan untuk membentuk individu-individu baru yang digambarkan sebagai susunan gen (kromosom). Kromosom tersebut direpresentasikan sebagai alternatif solusi yang disediakan untuk menyelesaikan permasalahan yang ada. Kemudian dilanjutkan dengan proses reproduksi, yang bertujuan untuk mendapatkan keturunan (*offspring*) yang berasal dari individu-individu yang termasuk populasi. Proses selanjutnya adalah proses evaluasi dimana bertujuan untuk mengukur tingkat kebugaran (*fitness*) dari masing-masing kromosom. Tinggi nilai *fitness* berbanding lurus dengan peluang individu tersebut mempengaruhi peluang individu tersebut menjadi calon solusi. Lalu dilanjutkan dengan proses seleksi, yang bertujuan untuk memilih individu yang berasal dari populasi dan juga merupakan *offspring* yang mana dipertahankan untuk proses regenerasi selanjutnya. Individu yang memiliki nilai *fitness* yang tinggi memiliki peluang besar terpilih sebagai calon solusi (Gen & Cheng, 1997).

Algoritme genetika membutuhkan beberapa variabel atau parameter, antara lain adalah (Sihombing, 2014):

1. Fungsi *fitness* yang dimiliki oleh masing-masing individu, yang berguna dalam penentuan tingkat kesesuaian individu tersebut dengan kriteria yang diinginkan.
2. Populasi jumlah individu yang digunakan pada setiap generasi.
3. Probabilitas terjadinya proses persilangan (*crossover*) pada suatu generasi.
4. Probabilitas terjadinya proses mutasi pada setiap individu.
5. Jumlah generasi yang akan dibentuk, yang bertujuan untuk mengetahui lama penerapan algoritme genetika.

2.3.1 Struktur algoritme genetika

Dalam memecahkan suatu permasalahan, algoritme genetika memulai proses dengan melakukan pemetaan (*encoding*) pada solusi dari suatu masalah menjadi *chromosome*. *Chromosome* tersebut terdiri dari gen-gen yang menggambarkan variabel-variabel keputusan yang digunakan dalam solusi. Fungsi *fitness* dan representasi dari *chromosome* yang digunakan untuk menilai kualitas dan kelayakan suatu *chromosome* dimasukkan ke algoritme genetika. Beberapa kasus membuktikan bahwa kualitas dari solusi yang dihasilkan dipengaruhi oleh bagaimana cara merepresentasikan sebuah solusi menjadi *chromosome*. Alur pencarian solusi pada algoritme genetika ditunjukkan pada Gambar 2.1 (Mahmudy, 2013).



Gambar 0.1 Pencarian Solusi dengan Algoritme Genetika

Sumber : (Mahmudy, 2013)

Dalam melakukan proses pencarian solusi, Algoritme genetika dibagi menjadi beberapa proses, yaitu:

1. Inisialisasi

Proses inisialisasi bertujuan untuk membangkitkan himpunan alternatif solusi yang baru secara acak yang berisi beberapa *chromosome* dan diletakkan pada populasi. Pada tahap ini dilakukan proses penentuan ukuran populasi (*population size*). Selain itu juga dilakukan proses penentuan jenis representasi *chromosome* menggunakan teknik pengkodean (Mahmudy, 2013).

2. Reproduksi

Proses reproduksi bertujuan untuk menghasilkan individu baru hasil penurunan (*offspring*) dari individu lama yang ada di dalam populasi. Himpunan keturunan tersebut akan ditempatkan pada penampungan *offspring*. Terdapat dua metode yang dapat digunakan dalam proses reproduksi, yaitu tukar silang (*crossover*) dan mutasi (*mutation*) (Mahmudy, 2013).

a. Tukar Silang (*Crossover*)

Proses *crossover* bertujuan untuk mendapatkan solusi yang menuju konvergen pada suatu titik tertentu secara acak. Sebelumnya harus ditentukan terlebih dahulu nilai tingkat *crossover* (*crossover rate / cr*). *Crossover rate* menyatakan perbandingan *offspring* yang diperoleh dari hasil proses *crossover* terhadap ukuran populasi sehingga dihasilkan nilai *offspring* sebesar $pc \times population\ size$ (Mahmudy, 2013).

Pertama-tama tentukan terlebih dahulu nilai *cr*. Jika ditentukan nilai $cr=0,5$ dan *population size* sebanyak 4, maka akan terdapat $0,5 \times 4=2$ buah *offspring* yang akan dihasilkan dari proses tukar silang. Misal terdapat P1 dan P2 sebagai berikut.

$$P_1 \quad [0 \ 0 \ 1 \ 1]$$

$$P_3 \quad [1 \ 0 \ 0 \ 1]$$

Maka akan menghasilkan child C1 dan C2 sebagai berikut.



Proses tukar silang diatas menggunakan metode *one-cut-point crossover*, yaitu dengan cara menentukan terlebih dahulu titik potongnya, kemudian bit yang berada di belakang titik potong pada P1 ditukar dengan bit yang berada di belakang titik potong pada P3, dan juga sebaliknya.

b. Mutasi (*Mutation*)

Proses *mutation* bertujuan untuk mendapatkan individu baru sebagai keturunan sehingga individu baru tersebut menjadi heterogen yang mengubah suatu gen dari keturunan secara acak. Sebelumnya juga harus ditentukan terlebih dahulu nilai tingkat *mutation* (*mutation rate /*

pm). *Mutation rate* menyatakan perbandingan *offspring* yang diperoleh dari hasil proses *mutation* terhadap ukuran populasi sehingga dihasilkan nilai *offspring* sebesar $pm \times population\ size$ (Mahmudy, 2013).

Pertama-tama tentukan terlebih dahulu nilai pm . Jika ditentukan nilai $pc=0,2$ dan *population size* sebanyak 4, maka akan terdapat $0,2 \times 4=0,8$ (dibulatkan menjadi 1) buah *offspring* yang akan dihasilkan dari proses mutasi. Misal terdapat P_4 sebagai berikut.

$$P_4 \quad [0\ 1\ 0\ 1]$$

Maka akan menghasilkan child C_3 sebagai berikut.

$$P_4[0\ 1\ 0\ 1]$$



$$C_3 [0\ 1\ 0\ 0]$$

Proses mutasi di atas dilakukan dengan cara menentukan terlebih dahulu titik tukarnya. Kemudian nilai yang berada pada titik tukar diganti sesuai dengan algoritme yang digunakan. Untuk contoh kasus diatas, karena kromosomnya berbentuk bilangan biner, maka bit yang berada pada titik tukar (1) diganti menjadi sebaliknya (0).

3. Evaluasi

Proses evaluasi bertujuan untuk memperoleh nilai kebugaran (*fitness*) dari masing-masing *chromosome*. Semakin besar nilai *fitness* yang diperoleh dari setiap *chromosome*, maka semakin besar peluang *chromosome* tersebut menjadi kandidat solusi (Mahmudy, 2013). Nilai *fitness* merupakan nilai yang menggambarkan kualitas dari suatu individu (solusi). Nilai *fitness* tersebut digunakan sebagai pembeda kualitas dari *chromosome* dan dijadikan acuan untuk mencapai nilai yang optimal (Sihombing, 2014).

$$f = h \tag{0.5}$$

Keterangan :

f = nilai *fitness*

h = rumus fungsi maksimasi

Sedangkan untuk kasus ini menggunakan persamaan:

$$f = \frac{h}{5} \tag{0.6}$$

Perhitungan nilai *fitnes* pada optimasi menu makanan atlet berdasarkan jadwal latihan ini ditunjukkan dengan Persamaan (2.7) berikut:

$$f = \frac{(1 - norKal) + (1 - norKar) + (1 - norPro) + (1 - norLem) + (1 - norHar)}{5} \tag{0.7}$$



Nilai normalisasi yang digunakan dalam perhitungan *fitness* ini didapatkan dari nilai penalti (selisih) masing-masing kandungan gizi. Dimana nilai penalti dapat dihitung menggunakan Persamaan (2.8) sampai (2.11)

$$pinKal = |kebKal - kesKal| \quad (0.8)$$

$$pinKar = |kebKar - kesKar| \quad (0.9)$$

$$pinLem = |kebLem - kesLem| \quad (0.10)$$

$$pinPro = |kebPro - kesPro| \quad (0.11)$$

Kemudian untuk menghitung nilai normalisasi dapat menggunakan Persamaan (2.12) sampai (2.16)

$$norKal = \frac{pinKal}{domainKal} \quad (0.12)$$

$$norKar = \frac{pinKar}{domainKar} \quad (0.13)$$

$$norPro = \frac{pinPro}{domainPro} \quad (0.14)$$

$$norLem = \frac{pinLem}{domainLem} \quad (0.15)$$

$$norHar = \frac{totHarga}{domainHar} \quad (0.16)$$

Keterangan:

<i>f</i>	: <i>fitness</i>
kebKal	: kebutuhan aktual kalori
kebKar	: kebutuhan aktual karbohidrat
kebLem	: kebutuhan aktual lemak
kebPro	: kebutuhan aktual protein
kesKal	: kalori rekomendasi sistem
kesKar	: karbohidrat rekomendasi sistem
kesLem	: lemak rekomendasi sistem
kesPro	: protein rekomendasi sistem
pinKal	: penalti kalori
pinKar	: penalti karbohidrat
pinLem	: penalti lemak
pinPro	: penalti protein
norKal	: normalisasi kalori
norKar	: normalisasi karbohidrat
norPro	: normalisasi protein
norLem	: normalisasi lemak
norHar	: normalisasi harga
totHarga	: total harga
domainKal	: batas atas kalori
domainKar	: batas atas karbohidrat

domainPro : batas atas protein
 domainLem : batas atas lemak
 domainHar : batas atas harga

4. Seleksi

Proses seleksi bertujuan untuk mendapatkan individu yang diperoleh dari populasi dan *offspring* yang dipertahankan untuk generasi selanjutnya. Proses seleksi dilakukan agar memperoleh generasi yang lebih baik dari generasi sebelumnya. Semakin tinggi nilai *fitness* untuk setiap *chromosome*, maka semakin tinggi peluang *chromosome* tersebut terpilih (Mahmudy, 2013). Nilai *fitness* yang diperoleh akan digunakan untuk proses seleksi berikutnya. Setiap individu yang telah diseleksi mendapatkan probabilitas reproduksi yang dipengaruhi oleh nilai objektif dari individu tersebut terhadap nilai objektif dari individu lain (Sihombing, 2014). Terdapat beberapa metode seleksi yang sering digunakan antara lain adalah:

a. Elitism

Metode seleksi *elitism* bekerja dengan cara memilih individu dengan nilai *fitness* tertinggi dari kumpulan individu dalam populasi (parent dan offspring). Seleksi tersebut mengambil individu sejumlah *population size*.

b. Roulette Wheel

Metode seleksi menggunakan *Roulette Wheel* bekerja dengan cara menghitung nilai probabilitas seleksi yang dimiliki oleh masing-masing individu berdasarkan nilai *fitness*-nya. Probabilitas tersebut menggambarkan besarnya peluang yang dimiliki oleh suatu individu untuk terpilih. Kemudian dari nilai probabilitas tersebut dapat dihitung probabilitas kumulatif untuk proses seleksi individu.

Langkah pertama adalah menghitung total nilai *fitness* dari seluruh individu (parent dan offspring) dengan cara:

$$totalFitness = \sum_{k=1}^{popSize} fitness(P_k) \quad (0.17)$$

Selanjutnya menghitung nilai probabilitas seleksi untuk masing-masing individu dengan cara:

$$prob_k = \frac{fitness(P_k)}{totalFitness}, \quad k = 1, 2, \dots, popSize \quad (0.18)$$

Kemudian menghitung nilai probabilitas kumulatif untuk masing-masing individu dengan cara:

$$probCum_k = \sum_{j=1}^k prob_j, \quad k = 1, 2, \dots, popSize \quad (0.19)$$

Langkah terakhir adalah memutar *roulette wheel* untuk memilih individu dengan cara:

- Menentukan nilai r yang merupakan bilangan random dengan range antara $0 - 1$.
- Ketika nilai r tersebut mendekati nilai probabilitas dari suatu individu, maka individu tersebut yang terpilih.

c. *Replacement Selection*

Metode seleksi *Replacement Selection* bekerja dengan cara menggantikan induknya dengan offspringnya apabila nilai fitness pada offspring lebih besar daripada nilai fitness pada induk (Mahmudy, et al., 2013).

Misal terdapat populasi dengan *population size*=5 yang memiliki individu dengan himpunan offsping sebagai berikut:

Individu	Fitness
P1	10
P2	8
P3	4
P4	7
P5	6

Dari hasil reproduksi dihasilkan *child* sebagai berikut:

Individu	Parent	Fitness
C1	P2 dan P3	3
C2	P4 dan P5	8
C3	P3	5

Dengan menggunakan metode seleksi *Replacement Selection*, maka hasilnya menjadi sebagai berikut.

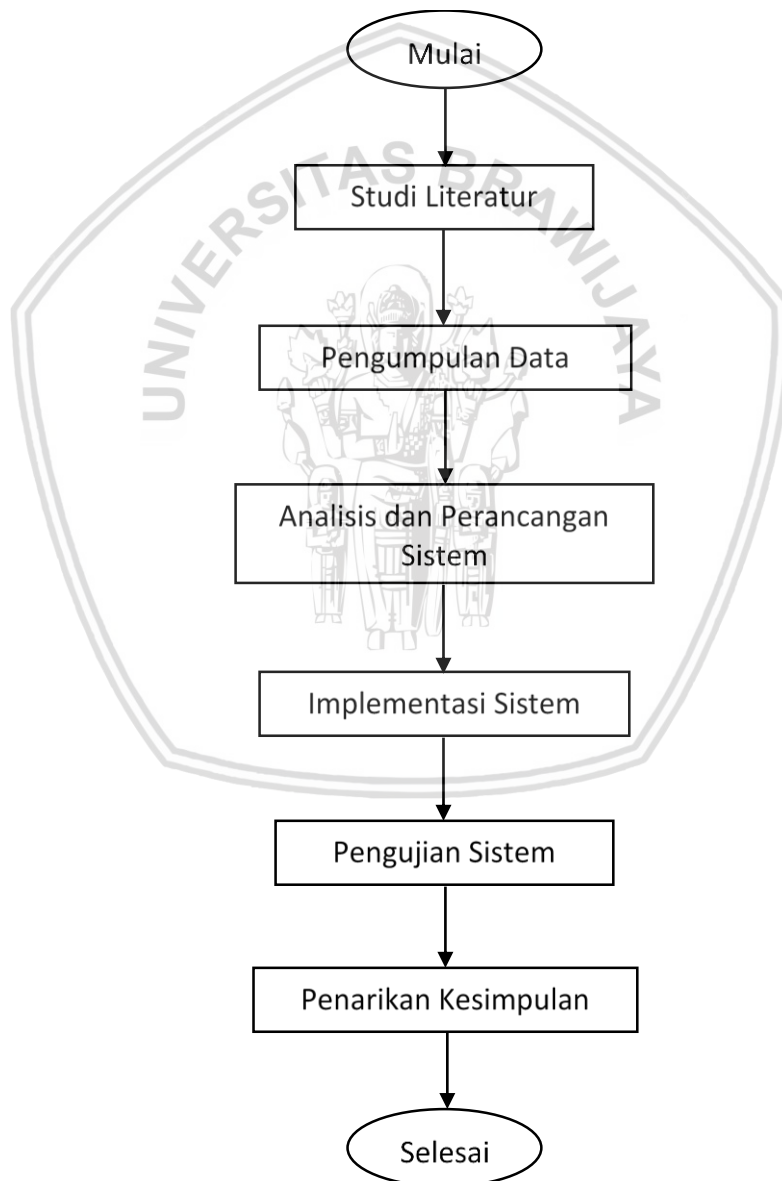
P(t+1)	Asal P(t)	Fitness
P1	P1	10
P2	P2	8
P3	C3	5
P4	P4	7
P5	C2	8

BAB 3 METODOLOGI

Pada bab ini membahas mengenai tahapan-tahapan yang akan dilakukan dalam penelitian pembuatan sistem Algoritme Genetika

3.1 Tahapan penelitian

Pada penelitian ini, tahapan penelitian yang dilakukan meliputi studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi sistem, pengujian sistem, dan penarikan kesimpulan. Tahapan penelitian tersebut dapat diilustrasikan dengan diagram blok seperti pada Gambar 0.1.



Gambar 0.1 Blog Diagram Metodologi Penelitian



3.1.1 Studi literatur

Pada tahap ini dilakukan pencarian literatur-literatur yang berkaitan untuk memperoleh informasi-informasi yang akan digunakan sebagai acuan dalam penelitian. Literatur tersebut menjelaskan dasar teori dan hasil dari penelitian sebelumnya untuk menunjang dalam penyusunan tugas akhir. Literatur yang diperoleh melingkupi:

- Kebutuhan Zat Gizi Atlet
- Algoritme Genetika

Pencarian Literatur dapat diperoleh dari buku atau jurnal penelitian yang telah dilakukan sebelumnya.

3.1.2 Pengumpulan data

Pada tahap ini dilakukan proses pengumpulan data yang dilakukan di berbagai *training center* yang ada seperti pemusatan atlet PON Tambora Provinsi Nusa Tenggara Barat, Unit Kegiatan Mahasiswa Universitas Brawijaya. Pengumpulan data dilakukan dengan mengambil data berupa jadwal latihan atlet yang sudah diterapkan dalam latihannya. Untuk data makanan dan kandungan gizi diambil dari data kandungan makanan yang didapatkan dari Dinas Kesehatan.

3.1.3 Analisis dan perancangan sistem

Analisis kebutuhan merupakan proses menganalisis kebutuhan yang bertujuan untuk mengidentifikasi kebutuhan yang diperlukan oleh sistem sehingga tidak menyimpang dari permasalahan dan tujuan penelitian. Analisis kebutuhan sistem meliputi:

1. Sistem mampu mengolah data atlet sehingga dapat menghitung kebutuhan energi atlet.
2. Sistem mampu melakukan proses dalam algoritme genetika yang kemudian menjadi sumber dalam menyusun menu makanan atlet.
3. Sistem dapat menampilkan data-data berupa data diri atlet, data latihan, data makanan, dan menu makanan yang direkomendasikan.

Perancangan sistem merupakan proses merancang sistem yang sesuai dengan penerapan untuk penerapan algoritme genetika untuk optimasi menu makanan atlet.

3.1.4 Implementasi sistem

Implementasi sistem merupakan proses pengaplikasian dari perancangan yang telah dibuat sebelumnya. Proses implementasi sistem akan dilakuakn dengan menggunakan bahasa pemrograman Java dan menggunakan database MySQL. Sistem akan dibuat dengan menerima masukan berupa parameter algoritme genetika. Masukan tersebut akan diproses dengan menggunakan algoritme genetika. Kemudian dari proses tersebut akan menghasilkan keluaran berupa

menu makanan yang dapat dimakan masing-masing atlet sesuai dengan jadwal latihan yang dilakukan.

3.1.5 Pengujian sistem

Pada tahap ini dilakukan pengujian yang bertujuan mengukur akurasi yang dihasilkan sistem. Pengujian akan dilakukan dengan 3 cara yaitu pengujian kombinasi nilai *Crossover Rate* (C_r) dan *Mutation Rate* (M_r), pengujian ukuran populasi, serta pengujian jumlah generasi.

3.1.6 Penarikan kesimpulan

Proses penarikan kesimpulan dilakukan setelah semua tahapan meliputi perancangan, implementasi, dan pengujian sistem telah selesai dilakukan. Tahap terakhir adalah pemberian saran yang bertujuan dalam pengembangan sistem selanjutnya serta untuk memperbaiki kesalahan yang terjadi pada penelitian ini.



BAB 4 PERANCANGAN

4.1 Formulasi permasalahan

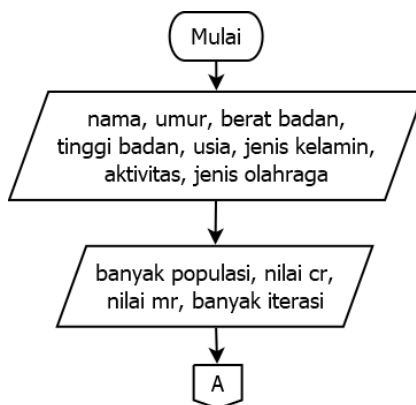
Permasalahan yang akan diselesaikan adalah optimasi menu makanan atlet berdasarkan jadwal latihan dengan menggunakan data makanan beserta kandungan gizinya. Data makanan yang digunakan terdiri dari 5 kategori yaitu sumber karbohidrat berjumlah 20 jenis, sumber protein hewani berjumlah 24 jenis, sumber protein nabati berjumlah 17 jenis, sayuran berjumlah 23 jenis, dan pelengkap (buah-buahan) berjumlah 23 jenis. Contoh data makanan sumber karbohidrat yang digunakan beserta kandungan gizinya dapat dilihat pada Tabel 4.1, untuk data lengkap yang digunakan dapat dilihat pada Lampiran B.

Tabel 0.1 Sumber Karbohidrat

No.	Nama Makanan	Berat (gr)	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)
1	Nasi Putih	180	234,0	51,5	4,3	0,4
2	Nasi Goreng	180	450,0	36,2	6,3	31,0
3	Nasi Jagung	180	217,7	49,0	5,0	1,1
4	Nasi Tim	180	210,8	46,3	4,0	0,4
5	Nasi Uduk	180	212,5	42,3	3,8	2,7
6	Bubur Nasi	180	131,2	28,8	2,3	0,2

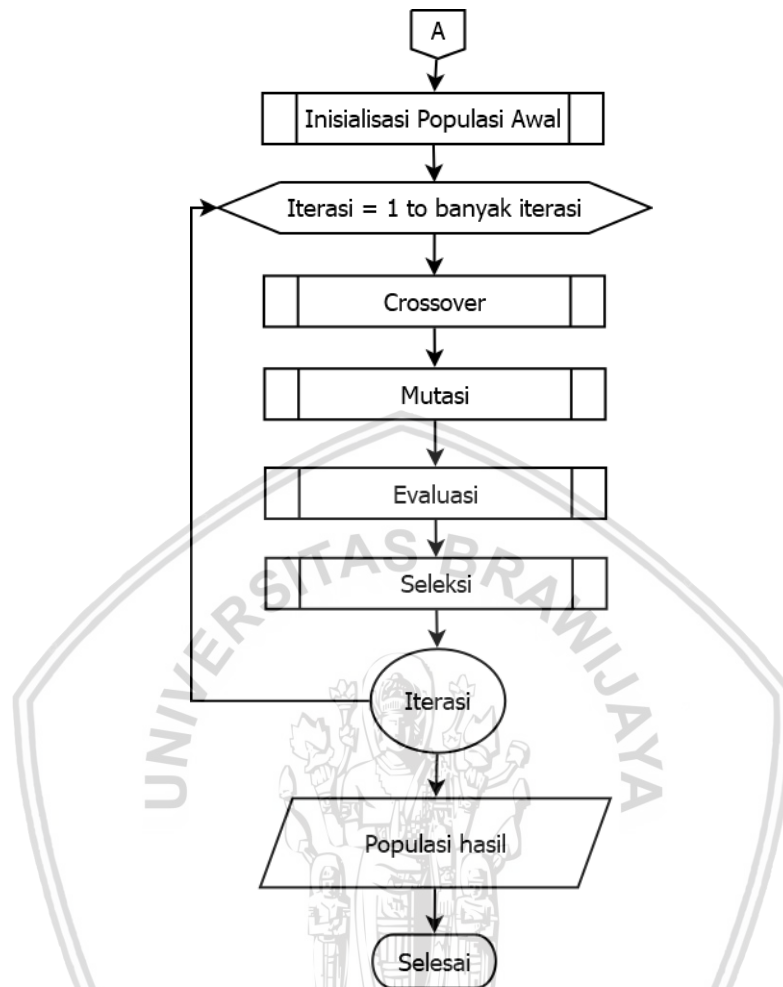
4.2 Tahapan algoritme genetika

Pada algoritme genetika terdapat beberapa tahapan yang dilakukan yaitu inialisasi populasi awal untuk mendapatkan individu-individu sebagai calon solusi, selanjutnya tahapan reproduksi yang terdiri dari *crossover* dan mutasi untuk menghasilkan keturunan baru, kemudian tahap evaluasi untuk menghitung nilai *fitness*, dan yang terakhir adalah seleksi untuk mendapatkan individu-individu pada generasi selanjutnya sekaligus individu terbaik. Berikut diagram alir algoritme genetika ditunjukkan pada Gambar 4.1 dan Gambar 4.2.



Gambar 0.1 Diagram Alir Algoritme Genetika

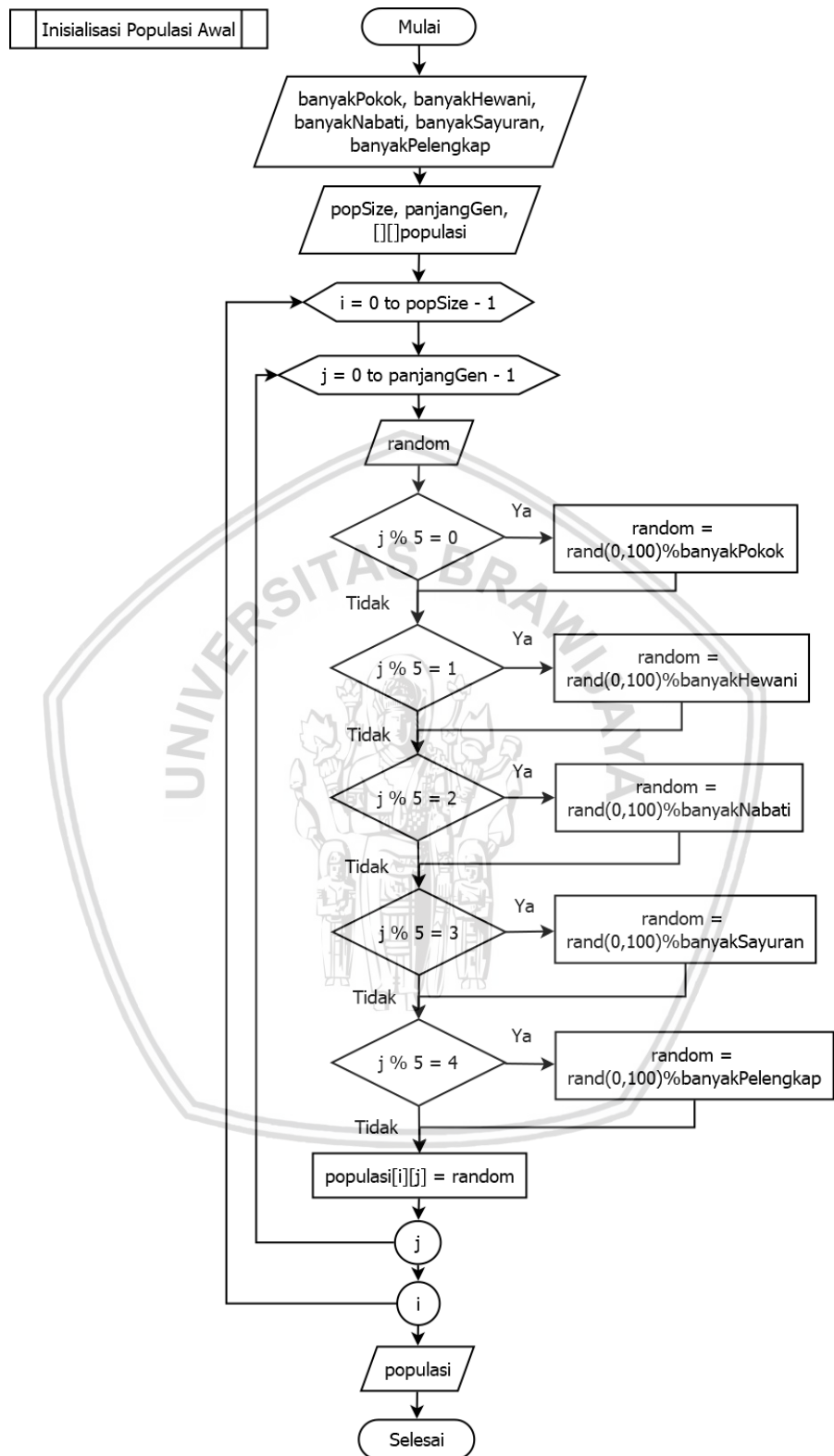




Gambar 0.2 Diagram Alir Algoritme Genetika (Lanjut)

4.2.1 Inisialisasi populasi awal

Tahap awal yang dilakukan adalah Inisialisasi populasi awal untuk mendapatkan individu-individu calon solusi sebanyak *population size*. Diawali dengan menentukan nilai *population size*, kemudian membangkitkan bilangan integer secara acak sebagai nilai gen yang merepresentasikan id makanan. Berikut diagram alir proses inisialisasi populasi awal pada permasalahan ini ditunjukkan pada Gambar 4.3.



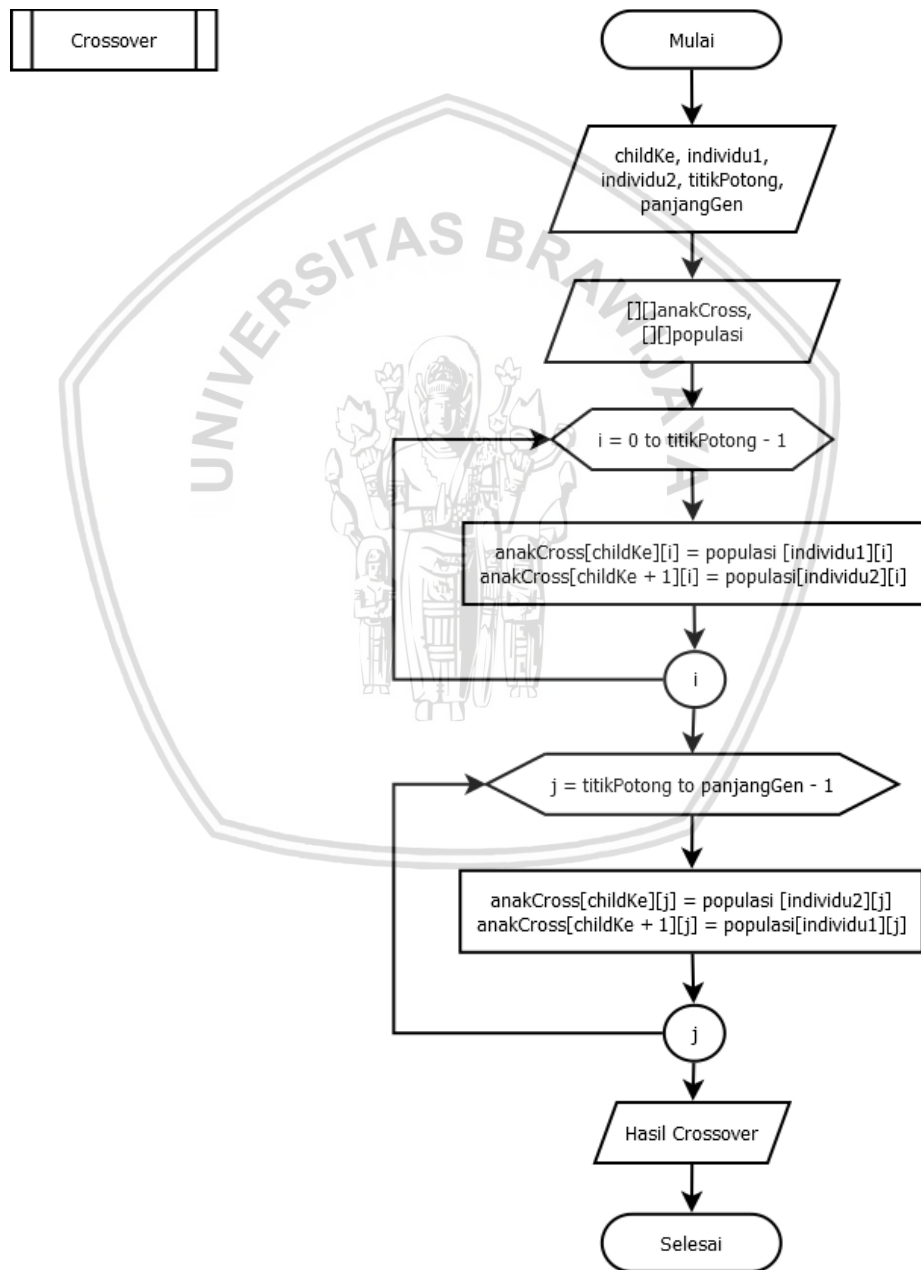
Gambar 0.3 Diagram Alir Inisialisasi Populasi Awal

Pada Gambar 4.3 proses inisialisasi populasi awal setiap gen dalam kromosom direpresentasikan dalam bilangan integer acak dengan batas bilangan maksimum

sama dengan jumlah sumber makanan yang direpresentasikan. Terdapat perulangan i yang merupakan perulangan untuk membangkitkan individu sebanyak *population size*, dan perulangan j merupakan perulangan untuk menentukan nilai setiap gen pada kromosom.

4.2.2 Crossover

Proses *crossover* pada penyelesaian permasalahan ini menggunakan metode *one cut point crossover*. Berikut diagram alir proses *crossover* ditunjukkan pada Gambar 4.4.

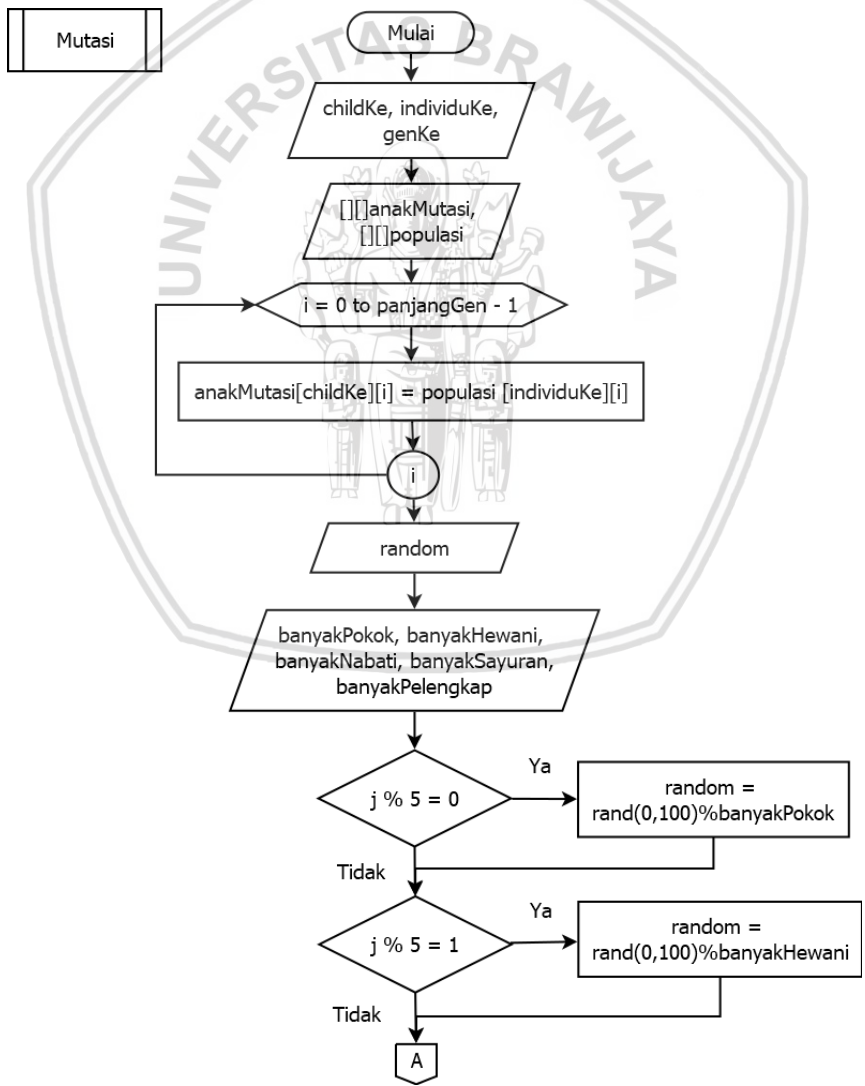


Gambar 0.4 Diagram Alir Proses *Crossover*

Gambar 4.4 menunjukkan proses *crossover*. Proses *crossover* diawali dengan menentukan menginputkan anak ke berapa yang akan dihasilkan, individu-individu mana saja yang akan menjadi *parent*, dan menentukan titik potong. Terdapat perulangan *i* sebanyak titik potong - 1, dimana dalam perulangan ini akan melakukan pengisian variabel *anakCross* dengan *anakCross* pertama disamakan dengan *individu1* dan *anakCross* kedua disamakan dengan *individu2*. Selanjutnya perulangan *j* yang dimulai dari titik potong sampai panjanggen - 1, dimana dalam perulangan ini akan melakukan pengisian variable *anakCross* dengan *anakCross* pertama disamakan dengan *individu2* dan *anakCross* kedua disamakan dengan *individu1*.

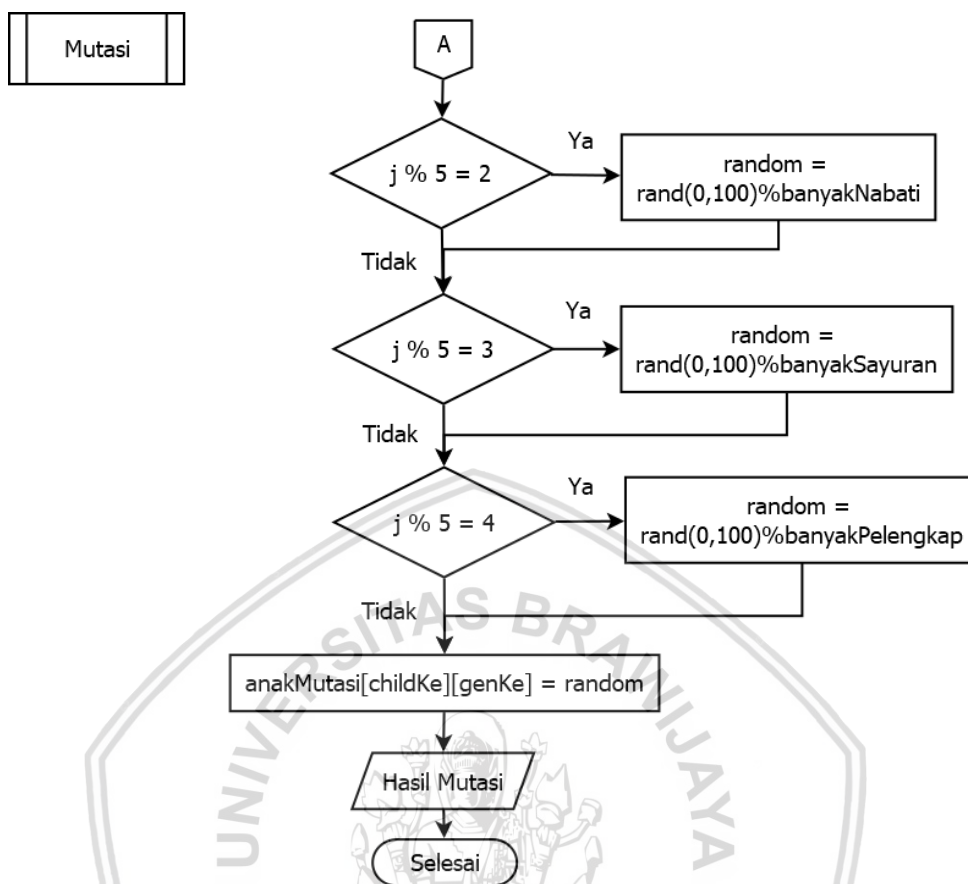
4.2.3 Mutasi

Metode mutasi yang digunakan adalah *random mutation*. Berikut diagram alir proses mutasi pada penelitian ini ditunjukkan oleh Gambar 4.5 dan Gambar 4.6.



Gambar 0.5 Diagram Alir Proses Mutasi



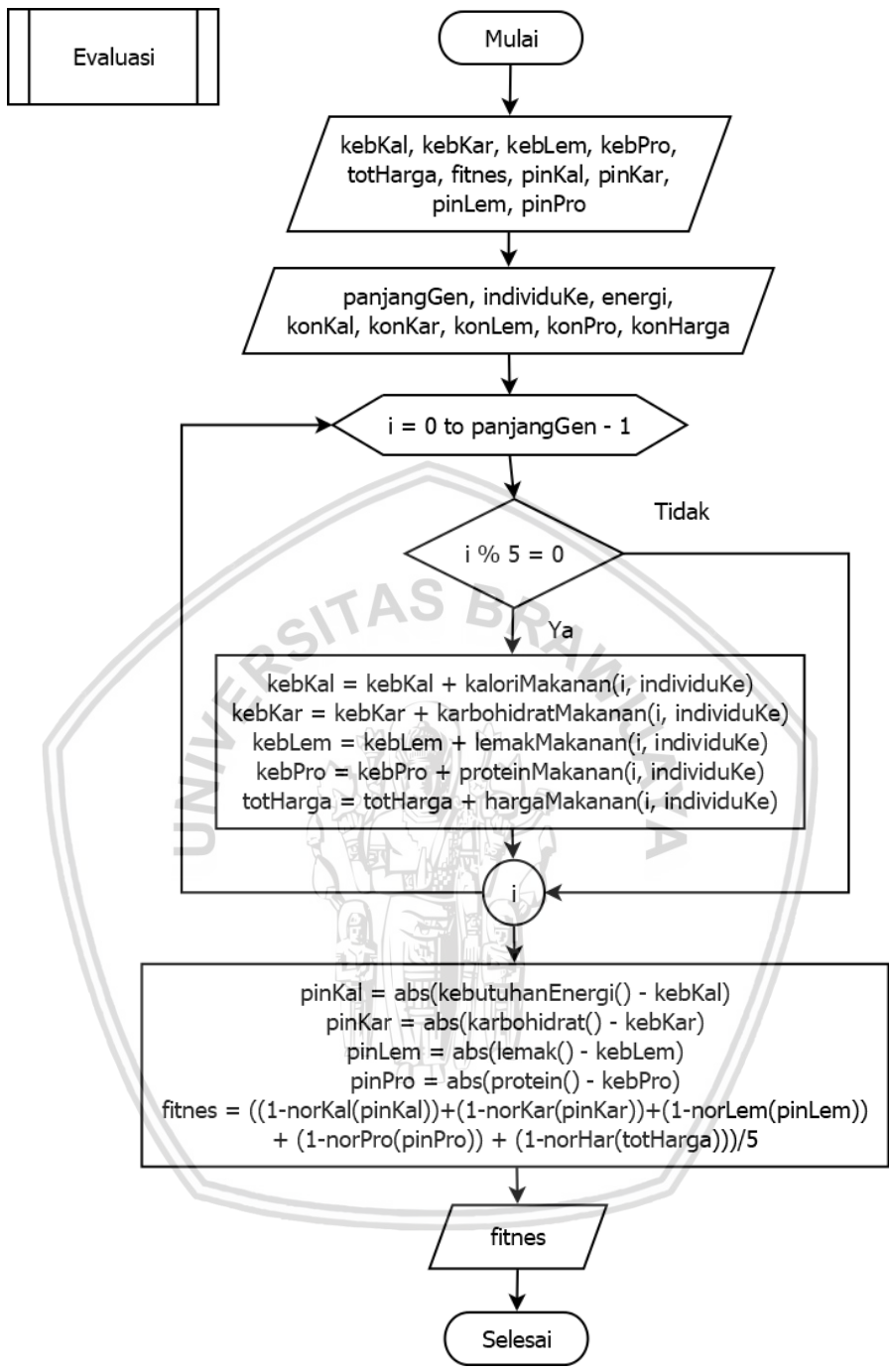


Gambar 0.6 Diagram Alir Proses Mutasi (Lanjutan)

Pada Gambar 4.5 dan Gambar 4.6 proses mutasi diawali dengan menentukan anak berapa yang akan dihasilkan, individu atau *parent* yang akan dimutasi, dan gen berapa yang akan dimutasi. Kemudian mendeklarasi anakMutasi sama dengan populasi ke (individu terpilih yang dimutasi). Selanjutnya melakukan proses mutasi pada gen yang akan dimutasi dengan cara yang sama dilakukan pada proses deklarasi individu random.

4.2.4 Evaluasi

Proses evaluasi dilakukan untuk menghitung nilai *fitness* masing-masing individu. Diagram alir proses evaluasi ditunjukkan oleh Gambar 4.7.

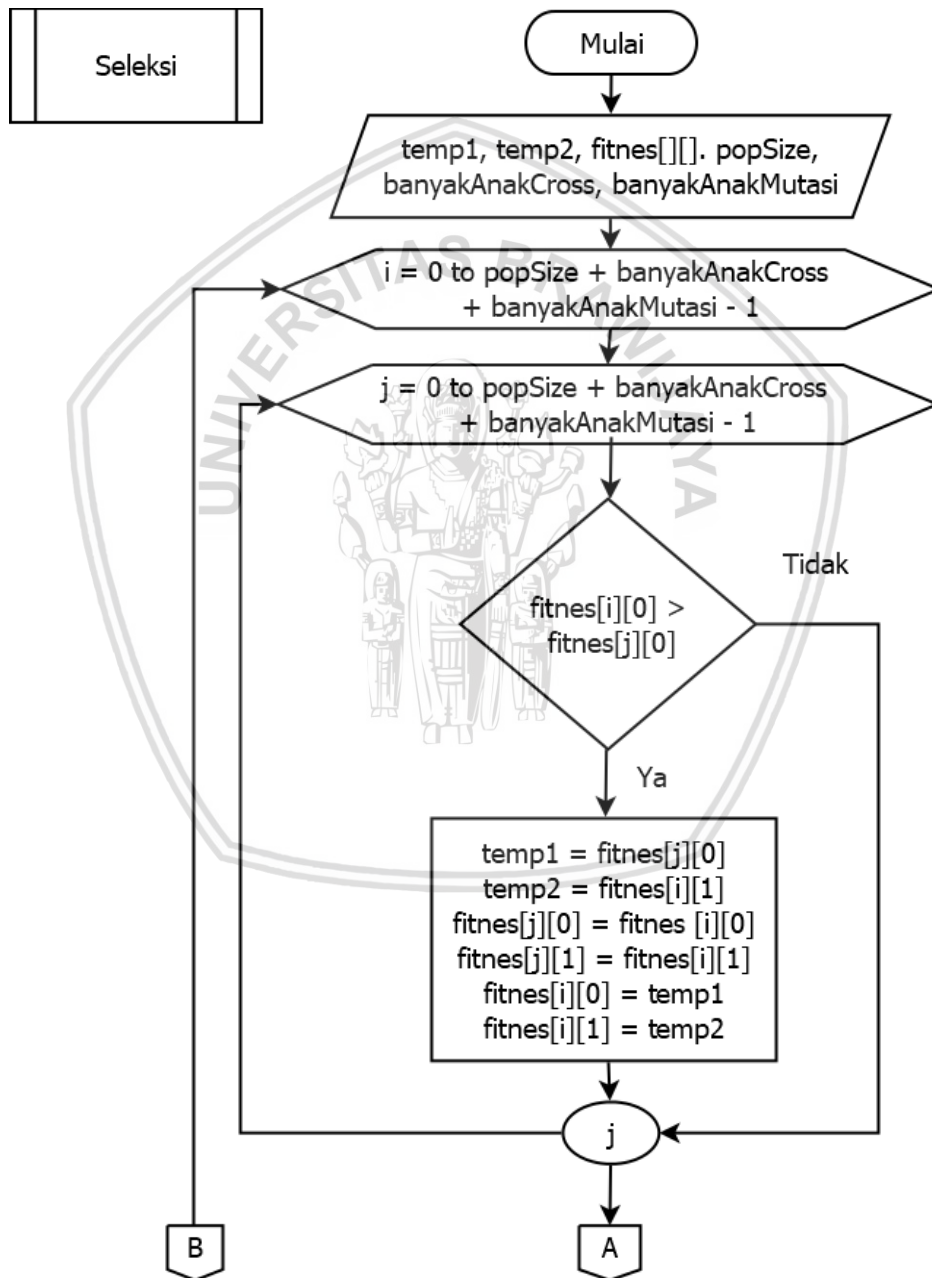


Gambar 0.7 Diagram Alir Evaluasi

Pada Gambar 4.7 proses evaluasi diawali dengan menentukan individu keberapa yang akan dihitung nilai fitnesnya. Kemudian dilakukan proses perhitungan besar kebutuhan masing-masing gizi makanan dan total harga berdasarkan gen yang dimiliki oleh individu tersebut. Setelah itu dilakukan proses perhitungan penalti masing-masing zat gizi dengan mencari selisih antara kebutuhan yang dibutuhkan dengan kebutuhan yang dihasilkan individu. Selanjutnya melakukan perhitungan nilai fitnes dengan membandingkan konstanta dengan penalti yang telah didapatkan.

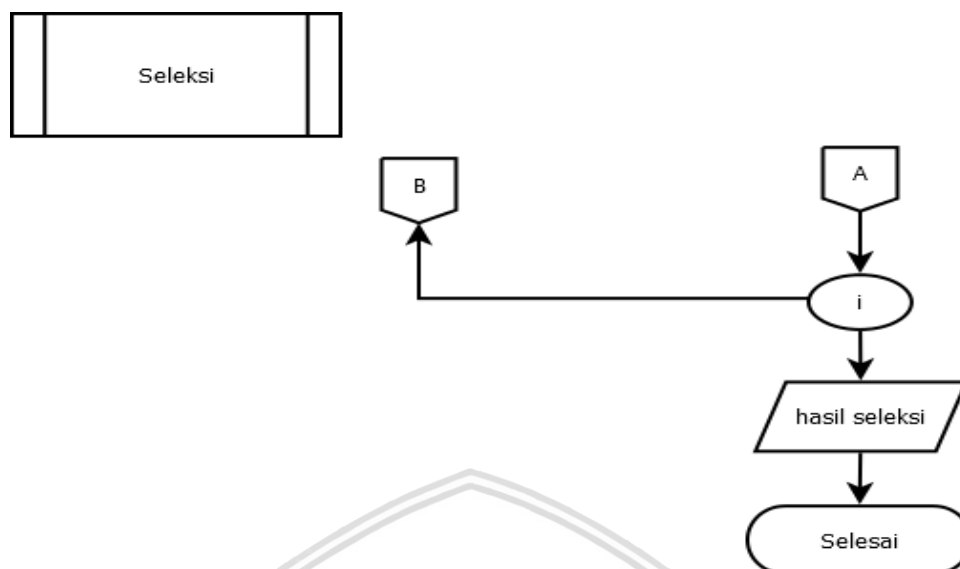
4.2.5 Seleksi

Proses seleksi yang dilakukan pada penyelesaian permasalahan ini menggunakan metode *elitism*. Awalnya populasi awal dan *offspring* hasil reproduksi dikumpulkan dalam satu penampungan, lalu mengurutkan berdasarkan nilai *fitness* tertinggi, dan mengambil individu dengan *fitness* terbaik sebanyak *population size*. Individu dengan nilai *fitness* tertinggi terpilih sebagai solusi. Berikut diagram alir proses seleksi *elitism* ditunjukkan oleh Gambar 4.8 dan Gambar 4.9.



Gambar 0.8 Diagram Alir Seleksi Elitism





Gambar 0.9 Diagram Alir Seleksi Elitism (Lanjut)

Pada Gambar 4.8 dan Gambar 4.9 proses seleksi diawali dengan mengurutkan nilai *fitness* setiap individu dari yang terbesar hingga terkecil. Kemudian dipilih individu dengan nilai *fitness* tertinggi sebanyak *population size* sebagai populasi baru dan individu dengan nilai *fitness* paling tinggi dipilih sebagai solusi.

4.3 Contoh perhitungan manual

4.3.1 Contoh permasalahan

Diberikan contoh permasalahan pembentukan menu makanan atlet dengan parameter *input* sebagai berikut:

- Nama : S
- Usia : 23 tahun
- Jenis Kelamin : Laki-laki
- Tinggi badan : 179 cm
- Berat badan : 76 kg
- Aktivitas : Aktif Ringan
- Cabang Olahraga : Atletik

Berdasarkan parameter *input* di atas maka dapat dihitung kebutuhan energi harian pada atlet tersebut dengan langkah pertama yang harus dilakukan adalah menghitung *basal metabolic rate* (BMR) menggunakan Persamaan (2.3).

$$BMR = 66,473 + 13,752 * 76 + 2,003 * 179 - 6,755 * 23$$

$$BMR = 1851,797$$



Berdasarkan hasil perhitungan *basal metabolic rate* atlet, selanjutnya dilakukan perhitungan *specific dynamic action* (SDA). Dimana SDA ini sendiri adalah 10% dari nilai BMR. Maka didapatkanlah nilai SDA sebesar:

$$SDA = 1851,797 * 10\%$$

$$SDA = 185,1797$$

Selanjutnya menentukan nilai aktivitas fisik, dimana nilai aktivitas fisik ini didapatkan berdasarkan Tabel 2.3. Berdasarkan tabel tersebut didapatkan nilai aktivitas fisik sebesar 1,11.

Setelah nilai BMR, SDA, dan aktivitas telah diketahui, selanjutnya dapat melakukan perhitungan kebutuhan energi harian menggunakan Persamaan (2.1)

$$Kebutuhan\ Energi\ Harian = (BMR + SDA) \times Aktivitas$$

$$Kebutuhan\ Energi\ Harian = (1851,797 + 185,1797) \times 1,11$$

$$Kebutuhan\ Energi\ Harian = 2837,74$$

Setelah mendapatkan nilai kebutuhan energi harian, selanjutnya melakukan perhitungan energi latihan. Perhitungan energi latihan ini didasarkan pada jadwal latihan. Dimana atlet yang terinput di atas adalah atlet cabang olahraga atletik, berikut merupakan jadwal latihan dari cabang olahraga atletik dapat dilihat pada Tabel 4.2.

Tabel 0.2 Jadwal Latihan Atletik

Senin	Selasa	Rabu	Kamis	Jumat
Beban	Koordinasi	Stabilisasi and Kondisioning	Teknik Dasar	Beban
2.6 x 5,40 kg	7 Gerakan. 30 m 3 x 3	Sit up 20 x 3	Gawang 5Gw x 5 x 3	7.6 x 5,40 kg
3.10 x 5,40 kg	Asselerasi 70m x 6	Beck Up 20 x 3	Gawang 5Gw x 5 x 2 1k	3.10 x 5,65 kg
7.6 x 5,44 kg		Phuss Up 20 x 3	Hopping, kijang 40m x 6	5.10 x 5,100 kg
1.6 x 5,144 kg		Beck Up 20 x 3	Asselerasi 60m x 5	1.6 x 5.144 kg
Gawang kecil		Bench Presh 10 x 3		8.4 x 30m
10Gw.10 x 3		Sit Up Samping 20 x 3		Asselerasi 80x6
Kijang 30m x 5		Naik tangga 3 x 3		
Akselerasi 80 x 6		Aselerasi 60m x 5		

Berdasarkan data latihan di atas, didapatkan kebutuhan energi latihan dengan cara menghitung energi yang dibutuhkan pada setiap jenis latihan, misalkan perhitungan energi pada hari kamis: Gawang 5Gw x 5 x 3 = 500; Gawang 5Gw x 5 x 2 1k = 570; Hopping, kijang 40m x 6 = 620; Asselerasi 60m x 5 = 410. Didapatkan total kebutuhan energi hari kamis = 2100. Dengan menggunakan cara

yang sama untuk setiap harinya, maka didapatkan kebutuhan energi latihan untuk setiap hari sebagai berikut: Senin = 2500, Selasa = 2210, Rabu = 2650, Kamis = 2100, Jumat = 2400.

Kemudian setelah menghitung kebutuhan energi harian dan energi latihan, maka langkah selanjutnya adalah menghitung kebutuhan energi total menggunakan Persamaan (2.2).

$$\begin{aligned} \text{Kebutuhan Energi Total} &= \text{Kebutuhan Energi Harian} \\ &+ \text{Pengeluaran Energi Latihan} \end{aligned}$$

$$\begin{aligned} \text{Kebutuhan Energi Total} &= 2437,7 \times 5 + 2500 + 2210 \\ &+ 2650 + 2100 + 2400 \end{aligned}$$

$$\text{Kebutuhan Energi Total} = 23165,22$$

Setelah mendapatkan kebutuhan energi total selama lima hari, selanjutnya melakukan perhitungan kebutuhan masing-masing zat gizi sebagai berikut:

- Kebutuhan karbohidrat = $49/400 \times 23165,22 = 2837,74$
- Kebutuhan lemak = $36/900 \times 23165,22 = 926,61$
- Kebutuhan protein = $15/400 \times 23165,22 = 868,70$

4.3.2 Inisialisasi parameter algoritme genetika

Pada algoritme genetika terdapat parameter-parameter yang harus didefinisikan terlebih dahulu antara lain, jumlah generasi, *population size*, *crossover rate* (*cr*), dan *mutation rate* (*mr*). Untuk menyelesaikan permasalahan pada contoh permasalahan ditentukan parameter algoritme genetika sebagai berikut:

- Jumlah generasi : 2
- *Population size* : 3
- *Crossover rate* : 0,4
- *Mutation rate* : 0,6

4.3.3 Representasi kromosom

Pada penelitian ini digunakan 107 makanan olahan yang dibagi menjadi 5 jenis yaitu sumber karbohidrat sebanyak 20 jenis, protein hewani sebanyak 24 jenis, protein nabati sebanyak 17 jenis, sayuran sebanyak 23 jenis dan pelengkap (buah-buahan) sebanyak 23 jenis. Representasi kromosom yang digunakan adalah representasi bilangan integer. Bilangan tersebut menunjukkan id makanan yang akan dikonsumsi.

Berdasarkan contoh permasalahan panjang kromosom yang digunakan sebesar 75 yang terbagi dalam 5 kali makan dalam satu hari yaitu makan pagi, makan siang dan makan malam. Setiap kelipatan 15 pada gen kromosom tersebut

menunjukkan hari, dan setiap 5 gen menunjukkan menu makanan yang dikonsumsi pada satu kali makan yang terdiri dari sumber karbohidrat, protein hewani, protein nabati, sayuran, dan pelengkap. Kemudian kromosom akan diisi dengan nilai acak, dimana nilai acak ini memiliki batas dalam setiap kelipatan 5 dari kromosom tersebut.

Misalkan pada 5 kromosom awal yang terdiri dari sumber karbohidrat, protein hewani, protein nabati, sayuran, dan pelengkap akan diisi. Dalam pengisiannya, gen yang mewakili sumber karbohidrat akan diisi dengan nilai acak rentang 1 – banyak makanan sumber karbohidrat yaitu 20, gen yang mewakili sumber protein hewani akan diisi dengan nilai acak rentang 1 – 24, gen yang mewakili sumber protein nabati akan diisi dengan nilai acak rentang 1-17, gen yang mewakili sumber sayuran akan diisi dengan nilai acak rentang 1-23, dan gen yang mewakili sumber pelengkap akan diisi dengan nilai acak rentang 1-23. Begitu pula untuk gen-gen selanjutnya.

Contoh representasi kromosom yang digunakan dalam menyelesaikan contoh permasalahan ditunjukkan pada Tabel 4.3.

Tabel 0.3 Representasi Kromosom

Individu	Hari Ke	Pagi					Siang					Malam				
		K	Ph	Pn	S	Pel	K	Ph	Pn	S	Pel	K	Ph	Pn	S	Pel
P1	1	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17
	2	6	17	1	18	14	7	1	4	6	18	15	24	2	9	2
	3	18	6	17	10	12	16	11	3	4	4	7	12	3	16	3
	4	11	23	2	5	12	9	16	17	4	19	7	5	7	20	4
	5	15	2	9	17	17	20	12	6	4	2	13	21	12	21	1
P2	1	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13
	2	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2
	3	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4
	4	9	7	4	23	19	3	7	13	17	14	5	7	6	5	13
	5	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1
P3	1	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3
	2	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20
	3	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7
	4	19	24	12	13	3	18	2	16	1	7	8	8	7	9	16
	5	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14

Representasi kromosom pada Tabel 4.3 dapat dikonversikan menjadi daftar menu makanan yang ditunjukkan pada Tabel 4.4.



Tabel 0.4 Konversi Kromosom Menjadi Manu Makanan

P1							
Hari ke-	Waktu Makan	Nama Makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Harga (Rp)
1	Pagi	Ubi Goreng	585,1	101,9	17,3	11,5	2000
		Telur Asin	136	6,3	13	6,2	2500
		Kacang Hijau Rebus	118,5	8,8	5,4	7,5	1200
		Tumis Terong	46,4	3,7	6,3	0,4	5300
		Pisang Kepok	158,2	40,4	2,4	0,4	2400
	Siang	Roti Tawar Manis	201,6	42,3	4,2	1,5	3500
		Ikan Asin Goreng	370	0,1	11,8	14,9	4000
		Semur Tahu	177	7,7	8,6	13,4	1000
		Ketimun Mentah	69,8	15,8	3,8	0,6	2000
		Alpukat	130	34	1	0,6	3200
	Malam	Nasi Goreng	201,6	42,3	4,2	1,5	3500
		Ikan Kembung Goreng	93,5	0,6	5,8	7,3	2000
		Tahu Goreng	50	4,3	4,8	2	1200
		Buncis Mentah	207,9	21	3,6	14	4000
		Strawberry	118,1	30,6	0,4	0,8	2000
2	Pagi	Nasi Putih	191,7	36,3	6,2	2,1	2000
		Telur Puyuh	370	0,1	11,8	14,9	4000
		Tempe Goreng	76	1,9	8,1	4,8	1000
		Sayur Nangka	30,1	4,2	4,6	0,4	3300
		Pisang Ambon	57,8	18,6	2,2	0,6	3000
	Siang	Ubi Jalar Kuning Kukus	201,8	47,3	4,3	0,2	1500
		Sate Ayam	80,5	0	9,5	14,2	4500
		Tahu Kukus	13,5	2,5	1,1	0,3	1500
		Sayur Bening	23,9	3,8	3	0,4	2500
		Semangka	164	42,6	0,4	0,8	1550
	Malam	Mie Rebus	253,8	50,9	8,6	1,3	2000
		Semur Daging Ayam	41,9	0	9,1	0,3	1500
		Tempe Bacem	76	1,9	8,1	4,8	1000
		Tumis Bayam Dan Sawi	69,8	15,8	3,8	0,6	2000
		Anggur	142	31,2	1,4	0,6	8000



Tab 4.4 Konversi Kromosom Menjadi Menu Makanan (Lanjut)

P1							
Hari ke-	Waktu Makan	Nama Makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Harga (Rp)
3	Pagi	Lontong	226,7	55,4	2	0,5	1500
		Ikan Mas Pepes	40,5	0	5,8	1,8	1500
		Lodeh Tahu	14,5	2,6	0,9	0,1	1500
		Tumis Bayam	28,2	4,6	1	1	5700
		Melon	64,1	14,4	1,2	0,8	3800
	Siang	Bakmie Goreng	450	36,2	6,3	31	5000
		Ikan Kakap Goreng	41,9	0	7,4	1,1	1500
		Tempe Oreg	118,5	8,8	5,4	7,5	1200
		Sayur Bayam Wortel	66	15	2,4	0,6	3500
		Arbei	76,5	16,6	1,2	0,4	3000
	Malam	Ubi Jalar Kuning Kukus	217,7	49	5	1,1	3500
		Ikan Asin Goreng	92,5	0,8	6,4	6,9	2000
		Tempe Oreg	82,9	4,8	4,9	5,6	2500
		Selada Mentah	37,8	8,2	2	0,2	6000
		Alpukat	118,1	30,6	0,4	0,8	6000
4	Pagi	Ubi Goreng	363,6	57,3	2,5	14,2	2500
		Ayam Rebus	28,4	0	1,8	2,3	5000
		Tempe Bacem	167,5	30,1	11,5	0,6	1300
		Sayur Bayam Jagung	55,9	3,2	3,4	4,2	2200
		Melon	184	46,8	2	1	2600
	Siang	Ubi Jalar Ungu Kukus	226,7	55,4	2	0,5	1500
		Telur Dadar	103,5	21,5	2,2	1	5000
		Lodeh Tahu	137	10,6	13,7	4,5	1500
		Sayur Bayam Wortel	187,9	2,4	2,4	20,2	4200
		Kelengkeng	130	34	1	0,6	4000
	Malam	Ubi Jalar Kuning Kukus	234	51,5	4,3	0,4	3000
		Ikan Mas Goreng	166	1,9	13,1	11,6	3500
		Semur Tahu	58	10,4	3,8	0,3	1000
		Bayam Merah	66	15	2,4	0,6	3500
		Arbei	104,2	25,6	1,4	0,8	2750



Tabel 4.4 Konversi Kromosom Menjadi Menu Makanan (Lanjut)

P1							
Hari ke-	Waktu Makan	Nama Makanan	Kalori (kkal)	Karbohidrat (gr)	Protein (gr)	Lemak (gr)	Harga (Rp)
5	Pagi	Mie Rebus	338,6	63,9	4	7,6	1500
		Soto Ayam	41,9	0	9,1	0,3	1500
		Oncom Goreng	16	2,8	1,1	0,1	1800
		Selada Air Mentah	37,8	8,2	2	0,2	6000
		Strawberry	164	42,6	0,4	0,8	1550
	Siang	Nasi Merah	234	51,5	4,3	0,4	3000
		Ikan Asin Goreng	40,5	0	5,8	1,8	1500
		Pepes Tahu	206	1,7	7,3	20,3	1300
		Sayur Bayam Wortel	69,8	15,8	3,8	0,6	2000
		Anggur	64,1	11	1,6	0,8	3500
	Malam	Roti Tawar Manis	217,7	49	5	1,1	3500
		Bebek Goreng	101	0	8	7,5	1500
		Sup Kacang Hijau	14,5	2,6	0,9	0,1	1500
		Brokoli Rebus	187,9	2,4	2,4	20,2	4200
		Apel	130	34	1	0,6	4000

4.3.4 Inisialisasi populasi awal

Berdasarkan contoh permasalahan ditentukan jumlah *population size* yaitu 3 dan jumlah hari adalah 5 sehingga panjang kromosom setiap individu adalah $3 \times 15 = 75$. Inisialisasi populasi awal pada contoh permasalahan ditunjukkan pada Tabel 4.5 dengan P1, P2 dan P3 merupakan individu-individu yang terbentuk dalam populasi awal.

Tabel 0.5 Inisialisasi Populasi Awal

Individu	Kromosom																
	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17	...	
P1	...	6	17	1	18	14	7	1	4	6	18	15	24	2	9	2	...
	...	18	6	17	10	12	16	11	3	4	4	7	12	3	16	3	...
	...	11	23	2	5	12	9	16	17	4	19	7	5	7	20	4	...
	...	15	2	9	17	17	20	12	6	4	2	13	21	12	21	1	...
	...	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...
P2	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...



Tabel 0.6 Inisialisasi Populasi Awal (Lanjut)

Individu	Kromosom																
P3	...	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3	...
	...	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20	...
	...	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7	...
	...	19	24	12	13	3	18	2	16	1	7	8	8	7	9	16	...
	...	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14	...

4.3.5 Crossover

Proses *crossover* pada permasalahan ini menggunakan *one cut point crossover*. Awalnya ditentukan terlebih dahulu jumlah *offspring* yang akan dihasilkan dengan cara mengalikan nilai *crossover rate* dengan *population size*. Pada permasalahan ini dapat dihitung nilai *offspring* sebagai berikut:

$$offspring = 0,4 \times 3 = 1,2 \text{ (dibulatkan menjadi 2)}$$

Berdasarkan perhitungan di atas proses *crossover* akan menghasilkan 2 *offspring* sehingga hanya memerlukan satu kali proses *crossover*. Langkah selanjutnya yaitu memilih dua induk dalam populasi awal secara acak. Dimisalkan 2 induk yang dipilih yaitu P2 dan P3, setelah induk terpilih kemudian menentukan titik potong *crossover* yang akan dilakukan secara acak. Dimana disini titik potong yang terpilih misalkan adalah 60, selanjutnya dilakukanlah proses penukaran komposisi gen pada P2 dan P3 di gen ke 60. Proses *crossover* dapat dilihat pada Tabel 4.7.

Tabel 0.7 Proses Crossover

Individu	Kromosom																
P2		8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...
	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...
P3	...	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3	...
	...	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20	...
	...	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7	...
	...	19	24	12	13	3	18	2	16	1	7	8	8	7	9	16	...
	...	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14	...

Dari tabel tersebut proses yang selanjutnya dilakukan adalah dengan melakukan penukaran susunan gen dari P2 yang berwarna kuning digabungkan dengan gen P3 yang berwarna kuning, begitu juga gen P2 yang berwarna putih digabungkan dengan gen P3 yang berwarna putih.



Dari proses *crossover* didapatkan dua buah *offspring* yaitu C1 dan C2 yang dapat dilihat pada Tabel 4.8.

Tabel 0.8 Offspring Hasil Crossover

Individu	Kromosom																
C1	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...	
	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	16	...
	...	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14	...
C2	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3	...	
	...	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20	...
	...	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7	...
	...	19	24	12	13	3	18	2	16	1	7	8	8	7	9	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...

4.3.6 Mutasi

Metode mutasi yang digunakan yaitu *random mutation*. Proses ini diawali dengan menentukan jumlah *offspring* yang akan dihasilkan dengan cara mengalikan nilai *mutation rate* dengan *population size* sebagai berikut:

$$offspring = 0.6 \times 3 = 1.8 \text{ (dibulatkan menjadi 2)}$$

Berdasarkan hasil perhitungan offspring maka membutuhkan dua kali proses mutasi. Untuk setiap proses mutasi menggunakan satu induk. Langkah selanjutnya yaitu memilih induk dari populasi awal secara acak. Dimisalkan induk terpilih pada proses mutasi pertama adalah P1 dan mutasi kedua adalah P2. Selanjutnya adalah menentukan gen keberapa yang akan dimutasi, misalkan untuk P1 yang dimutasi adalah gen ke 30 dan pada P2 adalah gen ke 52. Proses mutasi dapat dilihat pada Tabel 4.9

Tabel 0.9 Proses Mutasi

Individu	Kromosom																
P1	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17	...	
	...	6	17	1	18	14	7	1	4	6	18	15	24	2	9	2	...
	...	18	6	17	10	12	16	11	3	4	4	7	12	3	16	3	...
	...	11	23	2	5	12	9	16	17	4	19	7	5	7	20	4	...
	...	15	2	9	17	17	20	12	6	4	2	13	21	12	21	1	...
P2	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...	
	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...



Selanjutnya gen yang terpilih akan dilakukan proses mutasi yaitu penggantian dengan angka random, dimana angka random yang digunakan disesuaikan dengan posisi gen tersebut berada pada gen keberapa dalam kromosom. Dimana posisi ini nanti akan menentukan nilai random yang akan dihasilkan.

Dari proses mutasi didapatkan dua *offspring* yaitu C3 dan C4 yang dapat dilihat pada Tabel 4.10.

Tabel 0.10 Offspring Hasil Mutasi

Individu	Kromosom																
	C3	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17	...
...		6	17	1	18	14	7	1	4	6	18	15	24	2	9	14	...
...		18	6	17	10	12	16	11	3	4	4	7	12	3	16	3	...
...		11	23	2	5	12	9	16	17	4	19	7	5	7	20	4	...
...		15	2	9	17	17	20	12	6	4	2	13	21	12	21	1	...
C4	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...	
	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	22	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...

4.3.7 Evaluasi

Setelah melalui proses reproduksi yaitu *crossover* dan mutasi didapatkan populasi gabungan yaitu populasi awal dan populasi offspring hasil *crossover* dan mutasi seperti ditunjukkan pada Tabel 4.11.

Tabel 0.11 Populasi Baru

Individu	Kromosom																
	P1	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17	...
...		6	17	1	18	14	7	1	4	6	18	15	24	2	9	2	...
...		18	6	17	10	12	16	11	3	4	4	7	12	3	16	3	...
...		11	23	2	5	12	9	16	17	4	19	7	5	7	20	4	...
...		15	2	9	17	17	20	12	6	4	2	13	21	12	21	1	...
P2	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...	
	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...
P3	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3	...	
	...	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20	...
	...	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7	...
	...	19	24	12	13	3	18	2	16	1	7	8	8	7	9	16	...
	...	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14	...



Tabel 0.12 Populasi Baru (Lanjut)

Individu	Kromosom																
	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...	
C1	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	7	13	17	14	5	7	6	5	16	...
	...	12	12	6	8	15	17	23	6	16	17	17	8	5	2	14	...
	...	12	22	8	19	20	17	13	5	19	1	15	4	11	9	3	...
C2	...	10	22	1	19	1	11	17	2	7	16	9	24	5	22	20	...
	...	19	15	2	23	18	11	7	17	18	9	3	2	7	22	7	...
	...	19	24	12	13	3	18	2	16	1	7	8	8	7	9	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...
	...	11	14	11	14	16	13	12	7	22	3	2	4	5	15	17	...
C3	...	6	17	1	18	14	7	1	4	6	18	15	24	2	9	14	...
	...	18	6	17	10	12	16	11	3	4	4	7	12	3	16	3	...
	...	11	23	2	5	12	9	16	17	4	19	7	5	7	20	4	...
	...	15	2	9	17	17	20	12	6	4	2	13	21	12	21	1	...
	...	8	11	4	11	13	4	22	7	21	6	8	18	10	1	13	...
C4	...	16	15	6	8	13	18	24	1	18	14	6	7	9	12	2	...
	...	9	21	13	5	15	19	15	1	18	13	11	13	16	22	4	...
	...	9	7	4	23	19	3	22	13	17	14	5	7	6	5	13	...
	...	10	3	12	3	3	1	7	13	7	13	19	23	5	5	1	...

Pada proses evaluasi akan dihitung nilai *fitness* masing-masing individu dalam populasi untuk menentukan kualitas masing-masing individu. Sebelum menghitung nilai *fitness* ada beberapa yang dihitung terlebih dahulu yaitu kandungan zat yang dimiliki oleh masing-masing individu berdasarkan kromosom yang dimiliki dan menghitung penalti.

4.3.7.1 Perhitungan kandungan zat

Setelah melalui proses *crossover* dan mutasi populasi akan bertambah dengan offspring hasil *crossover* dan mutasi seperti ditunjukkan pada Tabel 4.11 dan 4.12.

Langkah pertama yang harus dilakukan adalah menghitung total gizi yaitu karbohidrat, protein, lemak, dan harga masing-masing individu. Jumlah kandungan gizi sistem ditunjukkan pada Tabel 4.13 dan Tabel 4.14.

Tabel 0.13 Jumlah Kandungan Gizi Makanan Sistem

Individu	Total				
	Kalori	Karbohidrat	Protein	Lemak	Harga
P1	10757,3	1440,21	405,41	395,61	227650
P2	11388,4	1529,1	461,9	362,8	204150
P3	11674,8	1563,9	505,6	348,6	207900
C1	11196,4	1601	410,9	370,4	211050



Tabel 0.14 Jumlah Kandungan Gizi Makanan Sistem (Lanjut)

Individu	Total				
	Kalori	Karbohidrat	Protein	Lemak	Harga
C2	11866,8	1492	556,6	341	224000
C3	11415,4	1533,9	460,3	367,4	227650
C4	10516,4	1449,4	414,8	366,6	211000

4.3.7.2 Perhitungan penalti

Berdasarkan jumlah kandungan gizi makanan pada Tabel 4.13 dan Tabel 4.13 dapat dihitung penalti masing-masing kebutuhan gizi menggunakan Persamaan (2.8) sampai (2.11) sebagai berikut:

- P1

$$\text{Penalti kalori} = |23165,22 - 10757,3| = 12407,92$$

$$\text{Penalti karbohidrat} = |2837,74 - 1440,21| = 1397,53$$

$$\text{Penalti protein} = |926,61 - 405,41| = 556,30$$

$$\text{Penalti lemak} = |868,70 - 395,61| = 472,29$$

Secara lengkap perhitungan penalti masing-masing individu dapat dilihat pada Tabel 4.15.

Tabel 0.15 Penalti Individu

Individu	Penalti				Total Harga
	Kalori	Karbohidrat	Lemak	Protein	
P1	12407,92	1397,53	521,20	473,09	227650
P2	12374,32	1341,63	556,30	472,29	204150
P3	11490,42	1273,84	421,01	520,10	207900
C1	11968,82	1236,74	515,71	498,30	211050
C2	11298,42	1345,74	370,01	527,70	224000
C3	11749,82	1303,84	466,31	501,30	227650
C4	12648,82	1388,34	511,81	502,10	211000

4.3.7.3 Perhitungan normalisasi

Berdasarkan nilai penalti yang telah didapatkan, kemudian dilakukan perubahan data menjadi data normalisasi pada masing-masing kandungan gizi menggunakan Persamaan (2.12) sampai (2.16) sebagai berikut:

- P1

$$\text{norKal} = \frac{12407,92}{22000} = 0,56399$$

$$\text{norKar} = \frac{1397,53}{3000} = 0,46584$$



$$norLem = \frac{521,20}{1100} = 0,47382$$

$$norPro = \frac{473,09}{1000} = 0,47309$$

$$norHar = \frac{227650}{260000} = 0,87557$$

Secara lengkap hasil normalisasi dapat dilihat pada Tabel 4.16.

Table 0.16 Normalisasi individu

Individu	Kalori	Karbohidrat	Protein	Lemak	Total Harga
P1	0,56399	0,46584	0,47309	0,47382	0,87557
P2	0,56246	0,44721	0,47229	0,50573	0,78519
P3	0,58965	0,46594	0,53539	0,49655	0,90615
C1	0,60704	0,47194	0,52339	0,53418	0,87865
C2	0,54660	0,45984	0,43618	0,48502	0,90307
C3	0,57107	0,46904	0,48249	0,48164	0,86480
C4	0,59029	0,46564	0,52959	0,50055	0,91192

4.3.7.4 Perhitungan *fitness*

Setelah mendapatkan penalti masing-masing gizi, selanjutnya menghitung nilai *fitness* masing-masing individu menggunakan Persamaan 2.7. Hasil perhitungan *fitness* dapat dilihat pada Tabel 4.17.

Fitness P1

$$= \frac{(1 - 0,56399) + (1 - 0,46584) + (1 - 0,47309) + (1 - 0,47382) + (1 - 0,87557)}{5}$$

$$= 0,42953$$

Tabel 0.17 Nilai *Fitness* Individu

Individu	Penalti				Total Harga	Fitness
	Kalori	Karbohidrat	Lemak	Protein		
P1	12407,92	1397,53	521,20	473,09	227650	0,42953
P2	11776,82	1308,64	464,71	505,90	227150	0,44541
P3	11490,42	1273,84	421,01	520,10	207900	0,40125
C1	11968,82	1236,74	515,71	498,30	211050	0,39695
C2	11298,42	1345,74	370,01	527,70	224000	0,43383
C3	11749,82	1303,84	466,31	501,30	227650	0,42618
C4	12648,82	1388,34	511,81	502,10	211000	0,40039



4.3.8 Seleksi

Proses seleksi pada penyelesaian permasalahan ini menggunakan metode seleksi *elitism*. Individu-individu dalam populasi diurutkan berdasarkan nilai *fitness* terbesar hingga terkecil dan individu sebanyak *population size* dengan nilai *fitness* terbesar akan terpilih. Hasil proses seleksi ditunjukkan pada Tabel 4.18

Tabel 0.18 Hasil Seleksi

Individu	Penalti				Total Harga	Fitness
	Kalori	Karbohidrat	Lemak	Protein		
P2	11776,82	1308,64	464,71	505,90	227150	0,44541
C2	11298,42	1345,74	370,01	527,70	224000	0,43383
P1	12407,92	1397,53	521,20	473,09	227650	0,42953

Populasi baru tersebut akan melalui tahap *crossover*, mutasi, evaluasi dan seleksi pada generasi ke-2. Hasil perhitungan *fitness* pada generasi ke-2 dapat dilihat pada Tabel 4.19

Tabel 0.19 Hasil Perhitungan *Fitness* Generasi Kedua

Individu	Penalti				Total Harga	Fitness
	Kalori	Karbohidrat	Lemak	Protein		
P1	11776,82	1308,64	464,71	505,90	227150	0,44541
P2	11298,42	1345,74	370,01	527,70	224000	0,43383
P3	12407,92	1397,53	521,20	473,09	227650	0,42953
C1	13029,72	1437,24	564,41	487,30	221100	0,43363
C2	11587,92	1187,84	463,11	513,10	200950	0,44562
C3	11778,82	1237,84	495,51	492,10	210550	0,43761
C4	11424,52	1257,64	420,81	518,50	204500	0,43040

Pada generasi ke-2 individu terbaik yang terpilih untuk dijadikan solusi adalah C2 dengan nilai *fitness* 0,44562

4.4 Perancangan antarmuka

Antarmuka digunakan sebagai perantara pengguna untuk dapat berinteraksi dengan sistem. Perancangan antarmuka pada sistem ini terdiri dari halaman *input data*, halaman algoritme genetika, halaman hasil, dan halaman lampiran.

4.4.1 Perancangan antarmuka halaman *input data*

Antarmuka halaman *input data* terdiri atas *input data* atlet dan *input operator algoritme* genetika. Perancangan halaman *input data* ditunjukkan pada Gambar 4.10 dan Gambar 4.11



SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran	
Data Atlet	Data Algen			

Nama

Umur Aktivitas

Tinggi Badan Jenis Kelamin

Berat Badan Jenis Olahraga

Data Atlet Data Algen

Gambar 0.10 Input Data Atlet

SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran	
Data Atlet	Data Algen			

Banyak Populasi

Nilai Cr

Nilai Mr

Banyak Iterasi

Data Atlet Data Algen

Gambar 0.11 Input Data Algoritma Genetika

4.4.2 Perancangan antarmuka halaman algoritme genetika

Halaman algoritme genetika akan menunjukkan insialisasi individu awal dan proses reproduksi yang dilakukan pada setiap iterasi. Perancangan halaman proses algoritme genetika ditunjukkan pada Gambar 4.12 dan Gambar 4.13.



SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran	
Inisialisasi	Reproduksi			

Inisialisasi Individu Awal

Data Atlet

Data Algen

Gambar 0.12 Halaman Algoritme Genetika Inisialisasi

SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran	
Inisialisasi	Reproduksi			

Iterasi	Individu	Fitness

Data Atlet

Data Algen

Gambar 0.13 Halaman Algoritme Genetika Reproduksi

4.4.3 Perancangan antramuka halaman hasil

Pada halaman hasil ini akan menunjukkan hasil berupa individu terbaik dan daftar rekomendasi menu makanan yang dapat dikonsumsi selama lima hari. Perancangan halaman hasil optimasi menu makanan ditunjukkan pada Gambar 4.14 dan Gambar 4.15.



SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran	
Individu Terbaik		Rekomendasi Menu Makanan		
<div style="border: 1px solid black; width: 100%; height: 100%; margin: 0 auto;"></div>				

Data Atlet

Data Algen

Gambar 0.14 Halaman Hasil Individu Terbaik

SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran																					
Individu Terbaik		Rekomendasi Menu Makanan																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Hari</th> <th style="width: 25%;">Pagi</th> <th style="width: 25%;">Siang</th> <th style="width: 25%;">Sore</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>					Hari	Pagi	Siang	Sore																
Hari	Pagi	Siang	Sore																					

Data Atlet

Data Algen

Gambar 0.15 Halaman Hasil Rekomendasi Menu Makanan

4.4.4 Perancangan halaman lampiran

Pada halaman ini akan menampilkan seluruh daftar makanan dan daftar latihan yang digunakan pada sistem ini. Perancangan halaman daftar makanan ditunjukkan pada Gambar 4.16 dan Gambar 4.17.

SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran
Data Makanan	Data Latihan		
Pokok	Hewani	Nabati	Sayuran
Pelengkap			

No	Nama Bahan	Berat	Kalori	Karbohidrat	Protein	Lemak	Harga

Data Atlet

Data Algen

Gambar 0.16 Halaman Lampiran Data Makanan

SISTEM OPTIMASI MENU MAKANAN ATLET

Input Data	Algoritme Genetika	Hasil	Lampiran
Data Makanan	Data Latihan		
Jenis Olahraga	Senin	Selasa	Rabu
Kamis			
Jumat			

Data Atlet

Data Algen

Gambar 0.17 Halaman Lampiran Data Latihan

4.5 Perancangan pengujian

Pengujian yang akan dilakukan terhadap sistem ini terdiri dari 3 pengujian antara lain:

1. Pegujian dengan kombinasi nilai *crossover rate* dan *mutation rate*
2. Pengujian populasi yaitu menentukan ukuran *population size* yang optimal
3. Pengujian generasi atau iterasi maksimum

4.5.1 Pengujian kombinasi *crossover rate* dan *mutation rate*

Pengujian kombinasi nilai *crossover rate* (cr) dan *mutation rate* menggunakan bilangan antara 0,1 hingga 0,9. Uji coba setiap kombinasi nilai cr dan mr dilakukan sebanyak 10 kali. Uji coba tersebut dapat dilihat pada Tabel 4.20.

Tabel 0.20 Pengujian *Crossover Rate* dan *Mutation Rate*

Kombinasi $C_r; M_r$	Nilai <i>Fitness</i> Pengujian Ke-										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0,1 ; 0,9											
0,2 ; 0,8											
0,3 ; 0,7											
0,4 ; 0,6											
0,5 ; 0,5											
0,6 ; 0,4											
0,7 ; 0,3											
0,8 ; 0,2											
0,9 ; 0,1											

4.5.2 Pengujian ukuran populasi

Pengujian populasi dilakukan dengan menentukan nilai *population size* yaitu banyaknya individu dalam populasi. Pengujian ini dilakukan untuk menentukan pengaruh ukuran populasi terhadap besarnya nilai *fitness* yang dihasilkan. Pengujian ukuran populasi pada sistem ini dilakukan sebanyak 10 kali dengan ukuran populasi kelipatan 10. Semakin besar ukuran populasi maka semakin banyak dan beragam individu yang dapat menghasilkan solusi terbaik. Uji coba ukuran populasi dapat dilihat pada Tabel 4.21.

Tabel 0.21 Pengujian Ukuran Populasi

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

4.5.3 Pengujian ukuran generasi

Pengujian ukuran generasi pada sistem ini dilakukan untuk mengetahui berapa ukuran generasi optimal yang dapat digunakan sisten untuk menghasilkan hasil optimal. Semakin tinggi ukuran dan kompleksitas masalah maka generasi semakin besar (Yogeswaran, Ponnambalam & Tiwari 2009). Pengujian generasi pada sistem ini dilakukan sebanyak 10 kali dengan ukuran generasi kelipatan 50. Uji coba generasi dapat dilihat pada Tabel 4.22.

Tabel 0.22 Pengujian Ukuran Generasi

Generasi	Nilai <i>Fitness</i> Pengujian Ke-										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50											
100											
150											
200											
250											
300											
350											
400											
450											
500											



BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem yang berdasarkan analisis kebutuhan serta proses perancangan sistem sebelumnya. Pada bab ini juga menjelaskan tentang implementasi dari algoritme genetika dan antarmuka yang digunakan dalam menyelesaikan permasalahan optimasi menu makanan atlet berdasarkan jadwal latihan.

5.1 Spesifikasi sistem

Dalam proses implementasi sistem terdapat spesifikasi perangkat yang dibutuhkan sesuai dengan kebutuhan dalam membangun sistem. Spesifikasi perangkat tersebut terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi perangkat keras

Spesifikasi perangkat keras yang digunakan dalam implementasi algoritme genetika untuk optimasi menu makanan atlet berdasarkan jadwal latihan ditunjukkan pada Tabel 5.1.

Tabel 0.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Prosesor	Processor Intel(R) Core(TM) i5-6400 CPU @2.70GHz
Memori	4.0 GB
Hardisk	500 GB

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi algoritme genetika untuk optimasi menu makanan atlet berdasarkan jadwal latihan ditunjukkan pada Tabel 5.2.

Tabel 0.2 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem operasi	Windows 10
Bahasa pemrograman	Java
Tools pemrograman	NetBeans IDE 8.2

5.2 Implementasi program

Implementasi program menjelaskan tentang bagaimana proses algoritme genetika dalam menyelesaikan masalah optimasi menu makanan atlet berdasarkan jadwal latihan dalam bentuk kode program.

5.2.1 Implementasi proses inialisasi populasi awal

Pada proses inialisasi populasi awal sistem mengacu pada perancangan diagram alir Gambar 4.3, dimana proses inialisasi populasi awal akan membangkitkan populasi secara acak sebanyak nilai *population size* yang telah dimasukkan. Panjang kromosom ditentukan oleh jumlah hari yang dimasukkan pengguna dikalikan dengan 15 yaitu hasil kali 5 jenis makanan dengan 3 kali makan dalam satu hari. Proses inialisasi populasi awal ditunjukkan pada Gambar 5.1.

setRandomIndividu	
1	public void setRandomIndividu() {
2	for (int i = 0; i < this.population size; i++) {
3	for (int j = 0; j < this.panjangGen; j++) {
4	int random = 0;
5	if (j % 5 == 0) {
6	random = (int) Math.ceil(Math.random() * banyakPokok);
7	}
8	if (j % 5 == 1) {
9	random = (int) Math.ceil(Math.random() * banyakHewani);
10	}
11	if (j % 5 == 2) {
12	random = (int) Math.ceil(Math.random() * banyakNabati);
13	}
14	if (j % 5 == 3) {
15	random = (int) Math.ceil(Math.random() * banyakSayuran);
16	}
17	if (j % 5 == 4) {
18	random = (int) Math.ceil(Math.random() * banyakPelengkap);
19	}
20	populasi[i][j] = random;
21	}
22	}
23	}

Gambar 0.1 Implementasi Kode Program Inialisasi Populasi Awal

Berikut penjelasan Kode Program Gambar 5.1:

1. Baris 1, proses inialisasi nama fungsi.
2. Baris 2-3, proses perulangan untuk pembangkitan populasi sebanyak *population size* dengan jumlah gen sebesar *panjangGen*.
3. Baris 4, proses menginisialisasi variabel *random* bernilai 0.
4. Baris 5-7, proses menentukan gen jika $\text{index} \bmod 5 = 0$ maka nilai *random* akan diisikan dengan gen jenis kebutuhan pokok.

5. Baris 8-10, proses menentukan gen jika index mod 5 = 1 maka nilai random akan diisikan dengan gen jenis protein hewani.
6. Baris 11-13, proses menentukan gen jika index mod 5 = 2 maka nilai random akan diisikan dengan gen jenis protein nabati.
7. Baris 12-16, proses menentukan gen jika index mod 5 = 3 maka nilai random akan diisikan dengan gen jenis sayuran.
8. Baris 17-19, proses menentukan gen jika index mod 5 = 4 maka nilai random akan diisikan dengan gen jenis pelengkap.
9. Baris 20, proses mengisikan nilai populasi dengan nilai random yang sudah diisikan sebelumnya.

5.2.2 Implementasi proses *crossover*

Proses implementasi *crossover* mengacu pada diagram alir Gambar 4.4, dimana proses ini diawali dengan menginputkan nilai parameter yang akan digunakan dalam menjalankan proses ini, adapun parameter dalam proses ini adalah *childKe*, *individu1*, *individu2*. Parameter ini berfungsi untuk menentukan anak keberapa yang akan dihasilkan dan individu keberapa saja yang akan digunakan sebagai *parent* dalam proses ini. Dalam satu kali proses *crossover* akan menghasilkan 2 individu baru. Proses *crossover* ditunjukkan pada Gambar 5.2.

crossOver(int childKe, int individu1, int individu2)	
1	public void crossOver(int childKe, int individu1, int individu2) {
2	int titikPotong = (int) Math.floor(Math.random() * this.panjangGen);
3	for (int i = 0; i < titikPotong; i++) {
4	anakCross[childKe][i] = populasi[individu1][i];
5	anakCross[childKe + 1][i] = populasi[individu2][i];
6	}
7	for (int j = titikPotong; j < this.panjangGen; j++) {
8	anakCross[childKe][j] = populasi[individu2][j];
9	anakCross[childKe + 1][j] = populasi[individu1][j];
10	}
11	}

Gambar 0.2 Implementasi Kode Program Proses *Crossover*

Berikut penjelasan Kode Program Gambar 5.2:

1. Baris 1, proses inialisasi nama fungsi dengan 3 parameter.
2. Baris2, proses inialisasi titik potong yang akan digunakan dalam *crossover*.
3. Baris 3-6, proses perulangan dari gen awal sampai titik potong - 1 untuk menampung anak hasil *crossover*.
4. Baris 7-10, proses perulangan dari titik potong sampai panjangGen -1 untuk menampung anak hasil *crossover*.



5.2.3 Implementasi proses mutasi

Proses reproduksi kedua yang dilakukan adalah proses mutasi. Pada proses ini pertama kali menginputkan parameter yang akan digunakan dalam prosesnya, yaitu *childKe*, *individuKe*, dan *genKe*. Parameter ini berfungsi untuk menentukan anak seberapa yang akan dihasilkan, indivi seberapa yang akan terpilih sebagai *parent*, dan gen seberapa yang akan dimutasi. Implementasi proses mutasi mengacu pada diagram alir yang terdapat pada Gambar 4.5, dimana hasilnya ditunjukkan pada Gambar 5.3.

mutasi (int childKe, int individuKe, int genKe)	
1	public void mutasi(int childKe, int individuKe, int genKe) {
2	for (int i = 0; i < this.panjangGen; i++) {
3	anakMutasi[childKe][i] = populasi[individuKe][i];
4	}
5	int random = 0;
6	if (genKe % 5 == 0) {
7	random = (int) Math.ceil(Math.random() * banyakPokok);
8	}
9	if (genKe % 5 == 1) {
10	random = (int) Math.ceil(Math.random() * banyakHewani);
11	}
12	if (genKe % 5 == 2) {
13	random = (int) Math.ceil(Math.random() * banyakNabati);
14	}
15	if (genKe % 5 == 3) {
16	random = (int) Math.ceil(Math.random() * banyakSayuran);
17	}
18	if (genKe % 5 == 4) {
19	random = (int) Math.ceil(Math.random() * banyakPelengkap);
20	}
21	anakMutasi[childKe][genKe] = random;
22	}

Gambar 0.3 Implementasi Kode Program Proses Mutasi

Berikut penjelasan Kode Program Gambar 5.3:

1. Baris 1, proses inialisasi fungsi dengan 3 parameter.
2. Baris 2, proses perulangan sebanyak panjangGen – 1.
3. Baris 3, proses dimana memberi nilai anak hasil mutasi sama dengan *parent*/in dividu terpilih.
4. Baris 5, proses deklarasi variabel random berniali 0.
5. Baris 6-8, proses menentukan gen jika index mod 5 = 0 maka nilai random akan diisikan dengan gen jenis kebutuhan pokok.

6. Baris 9-11, proses menentukan gen jika index mod 5 = 1 maka nilai random akan diisikan dengan gen jenis protein hewani.
7. Baris 12-14, proses menentukan gen jika index mod 5 = 2 maka nilai random akan diisikan dengan gen jenis protein nabati.
8. Baris 13-17, proses menentukan gen jika index mod 5 = 3 maka nilai random akan diisikan dengan gen jenis sayuran.
9. Baris 18-20, proses menentukan gen jika index mod 5 = 4 maka nilai random akan diisikan dengan gen jenis pelengkap.
10. Baris 21, proses mengisikan anak mutasi ke *childKe* dengan gen yang genKe dengan nilai random.

5.2.4 Implementasi proses perhitungan *fitness*

Proses perhitungan *fitness* dilakukan untuk menentukan kualitas setiap individu. Implementasi proses perhitungan *fitness* mengacu pada perancangan diagram alir yang terdapat pada Gambar 4.7, hasil implementasinya ditunjukkan pada Gambar 5.4.

hitungFitness(int individuKe) throws SQLException	
1	public double hitungFitnes(int individuKe) throws SQLException {
2	kebKal = 0;
3	kebKar = 0;
4	kebLem = 0;
5	kebPro = 0;
6	totHarga = 0;
7	double fitnes = 0;
8	double pinKal = 0, pinKar = 0, pinLem = 0, pinPro = 0;
9	for (int i = 1; i < panjangGen + 1; i++) {
10	if (i % 5 == 0) {
11	kebKal = kebKal + hitungKaloriMakanan(i - 4, individuKe);
12	kebKar = kebKar + hitungKarbohidratMakanan(i - 4, individuKe);
13	kebLem = kebLem + hitungLemakMakanan(i - 4, individuKe);
14	kebPro = kebPro + hitungProteinMakanan(i - 4, individuKe);
15	totHarga = totHarga + hitungHargaMakanan(i - 4, individuKe);
16	}
17	}
18	pinKal = Math.abs(energi.getKebuuthanEnergi() - kebKal);
19	pinKar = Math.abs(energi.getKarbohidrat() - kebKar);
20	pinLem = Math.abs(energi.getLemak() - kebLem);
21	pinPro = Math.abs(energi.getProtein() - kebPro);
22	fitnes = ((Math.abs(1 - dataNormalisasi[individuKe][0])) + (Math.abs(1 - dataNormalisasi[individuKe][1])) + (Math.abs(1 - dataNormalisasi[individuKe][2])) + (Math.abs(1 - dataNormalisasi[individuKe][3]))



23	+ (Math.abs(1 - dataNormalisasi[individuKe][4])) / 5;
24	return fitnes;
	}

Gambar 0.4 Implementasi Kode Program Proses Perhitungan *Fitness*

Berikut penjelasan Kode Program Gambar 5.4:

1. Baris 1, proses inialisasi fungsi dengan 1 parameter.
2. Baris 2-8, proses inialisasi variabel-variabel yang akan dipakai dalam proses ini.
3. Baris 9, proses perulangan sebanyak panjangGen - 1.
4. Baris 10-17, proses yang akan dilakukan jika index mod 5 = 0, maka akan melakukan proses perhitungan kebKal, kebKar, kebLem, kebPro, totHarga.
5. Baris 18-21, proses perhitungan nilai penalti yang akan digunakan dalam perhitungan fitnes.
6. Baris 22, proses perhitungan nilai *fitness*.

5.2.5 Implementasi proses seleksi

Proses seleksi dilakukan untuk menentukan individu-individu yang dipertahankan pada generasi selanjutnya. Implementasi proses seleksi mengacu pada diagram alir seleksi pada Gambar 4.8. Proses ini diawali dengan mengurutkan individu-individu berdasarkan nilai *fitness* dari tertinggi hingga terendah kemudian mengambil individu dengan nilai *fitness* tertinggi sejumlah *population size*. Implementasi proses seleksi ditunjukkan pada Gambar 5.5.

sortFitness()	
1	public void sortFitnes() {
2	double temp1, temp2;
3	for (int i = 0; i < population size + banyakAnakCross + banyakAnakMutasi; i++) {
4	for (int j = 0; j < population size + banyakAnakCross + banyakAnakMutasi; j++) {
5	if (this.fitnes[i][0] > fitnes[j][0]) {
6	temp1 = fitnes[j][0];
7	temp2 = fitnes[j][1];
8	fitnes[j][0] = fitnes[i][0];
9	fitnes[j][1] = fitnes[i][1];
10	fitnes[i][0] = temp1;
11	fitnes[i][1] = temp2;
12	}
13	}
14	}

Gambar 0.5 Implementasi Kode Program Proses Seleksi



Berikut penjelasan Kode Program Gambar 5.5 :

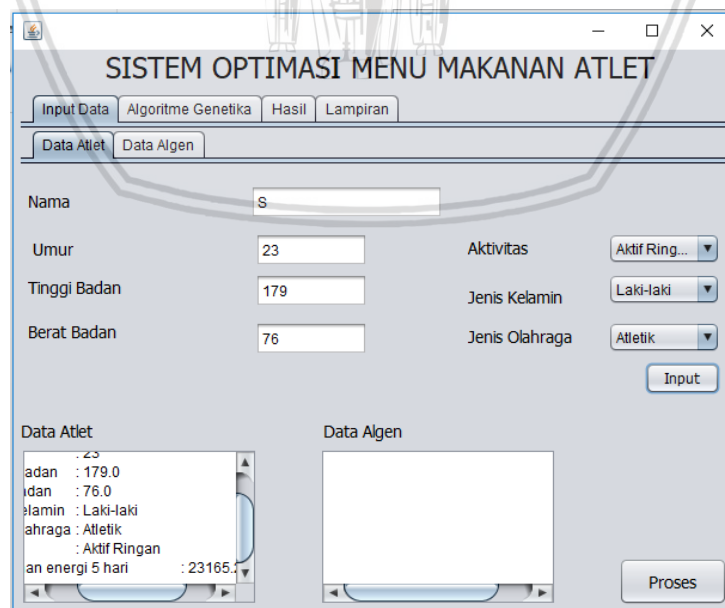
1. Baris 1, proses inisialisasi fungsi.
2. Baris 2, proses inisialisasi variabel temp1 dan temp2.
3. Baris 3, proses perulangan sebanyak individu yang ada (banyak *parent*, anak *cross*, dan anak mutasi) dengan index i.
4. Baris 4, proses perulangan sebanyak individu yang ada (banyak *parent*, anak *cross*, dan anak mutasi) dengan index j.
5. Baris 5-12, mengecek nilai fitness yang ada di index i apakah lebih besar dibandingkan index j, jika iya akan melakukan proses pengisian nilai fitness di index j dengan nilai fitness di index i.

5.3 Implementasi antarmuka

Implementasi antarmuka pada sistem ini terdiri dari halaman *input* data, halaman algoritme genetika, halaman hasil, dan halaman lampiran.

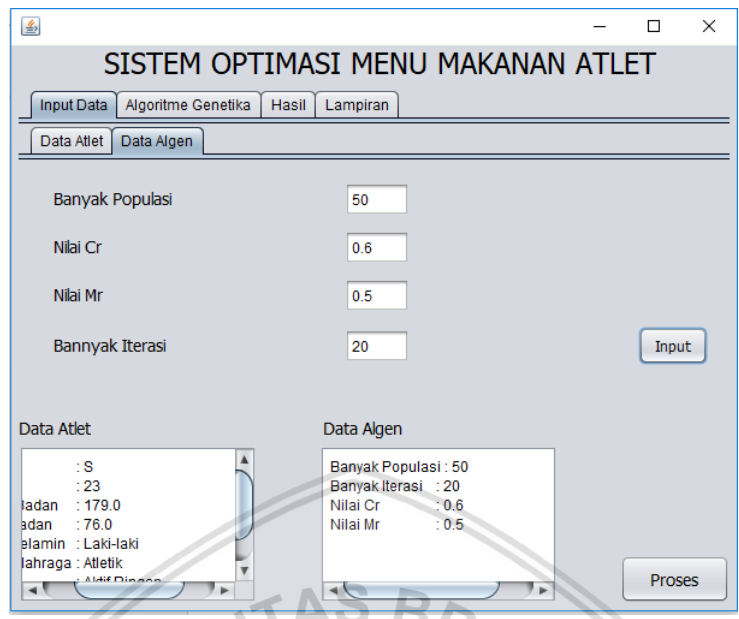
5.3.1 Implementasi antarmuka halaman *input* data

Pada implementasi antarmuka halaman *input* data terdiri atas *input* data atlet dan *input* operator *algoritme* genetika yang mengacu pada perancangan antarmuka Gambar 4.10 dan Gambar 4.11. Pada *input* data atlet akan ditampilkan operator-operator yang sudah diinputkan dan jumlah energi yang dibutuhkan oleh atlet tersebut, sedangkan untuk *input* data *algoritme* genetika akan menampilkan operator-operator yang sudah diinputkan sebelumnya. Implementasi antarmuka *input* data ditunjukkan pada Gambar 5.6 dan Gambar 5.7.



Gambar 0.6 Antarmuka Halaman *Input* Atlet

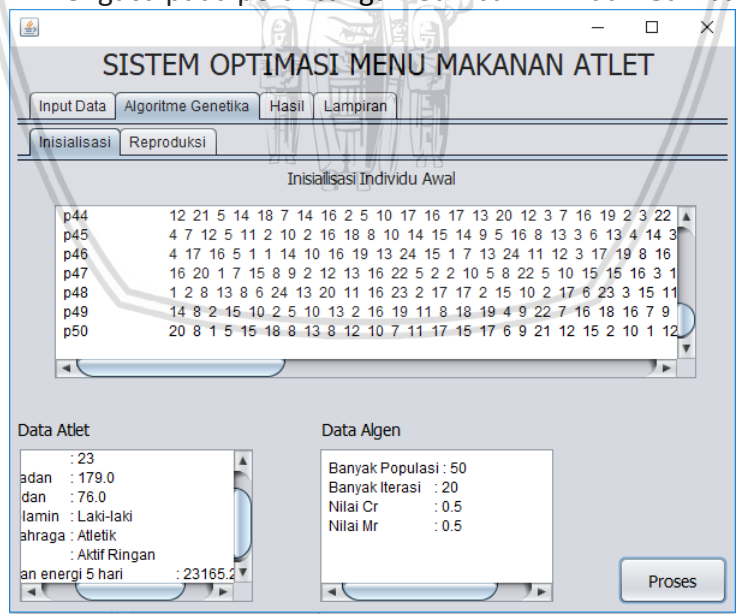




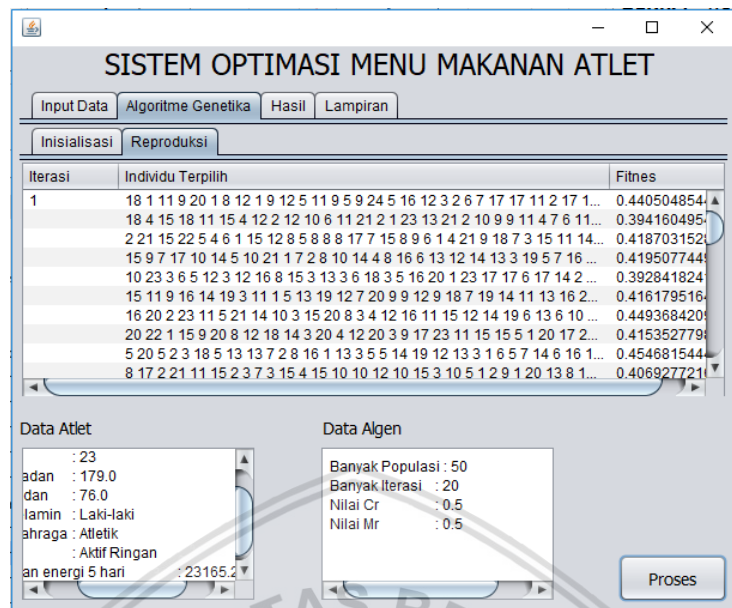
Gambar 0.7 Antarmuka Halaman *Input* Algoritme Genetika

5.3.2 Implementasi antarmuka halaman algoritme genetika

Halaman algoritme genetika akan menunjukkan insialisasi individu awal dan proses reproduksi yang dilakukan pada setiap iterasi. Halaman algoritme genetika ditunjukkan pada Gambar 5.8 dan Gambar 5.9., dimana implementasi antar muka ini mengacu pada perancangan Gambar 4.12 dan Gambar 4.13.



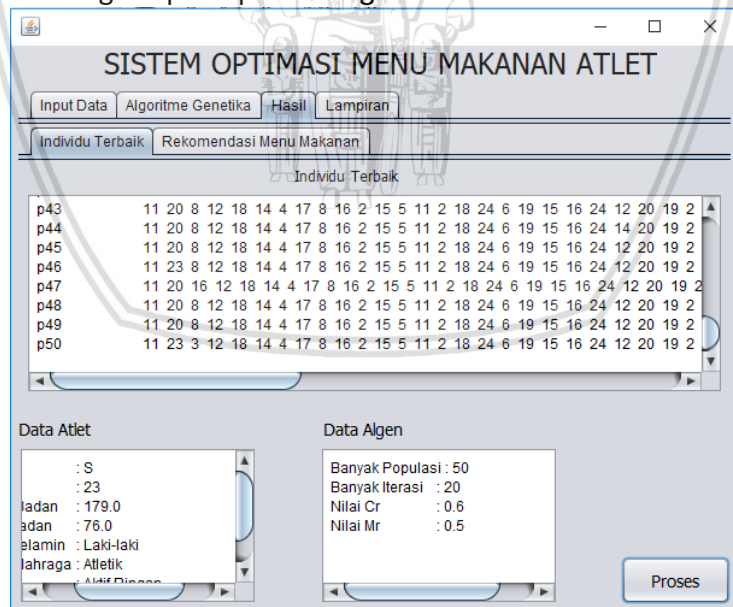
Gambar 0.8 Antarmuka Halaman Algoritme Genetika Inisialisasi



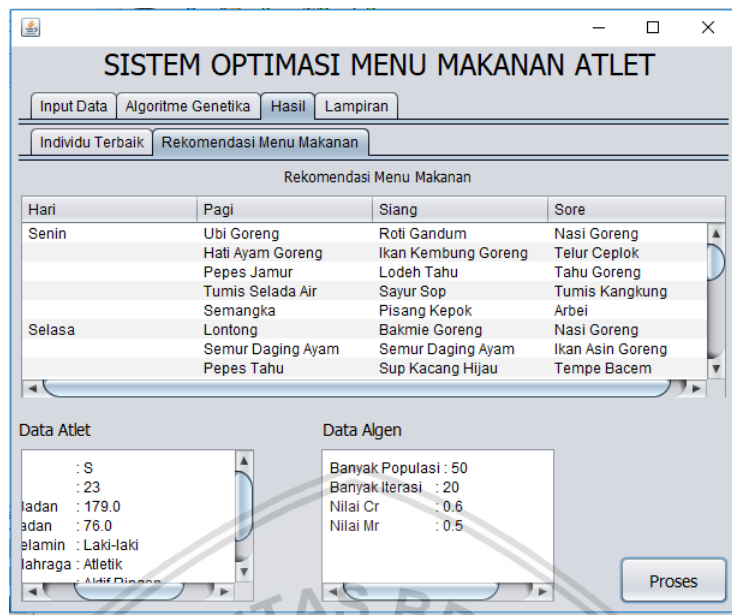
Gambar 0.9 Antarmuka Halaman Algoritme Genetika Reproduksi

5.3.3 Implementasi antarmuka halaman hasil

Halaman hasil ini akan menunjukkan hasil berupa individu terbaik dan daftar rekomendasi menu makanan yang dapat dikonsumsi selama lima hari. Halaman hasil dapat dilihat pada Gambar 5.10 dan Gambar 5.11., dimana implementasi antar muka ini mengacu pada perancangan Gambar 4.14 dan Gambar 4.15.



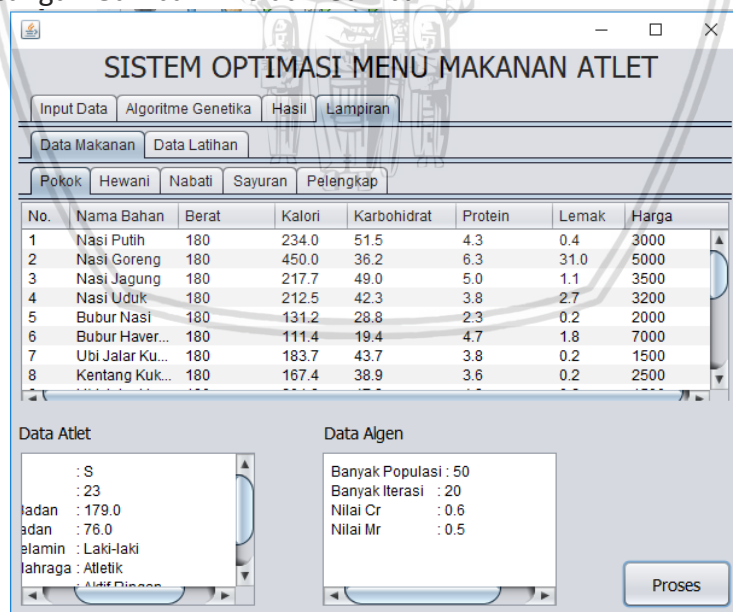
Gambar 0.10 Antarmuka Halaman Hasil Individu Terbaik



Gambar 0.11 Antarmuka Halaman Hasil Rekomendasi Menu Makanan

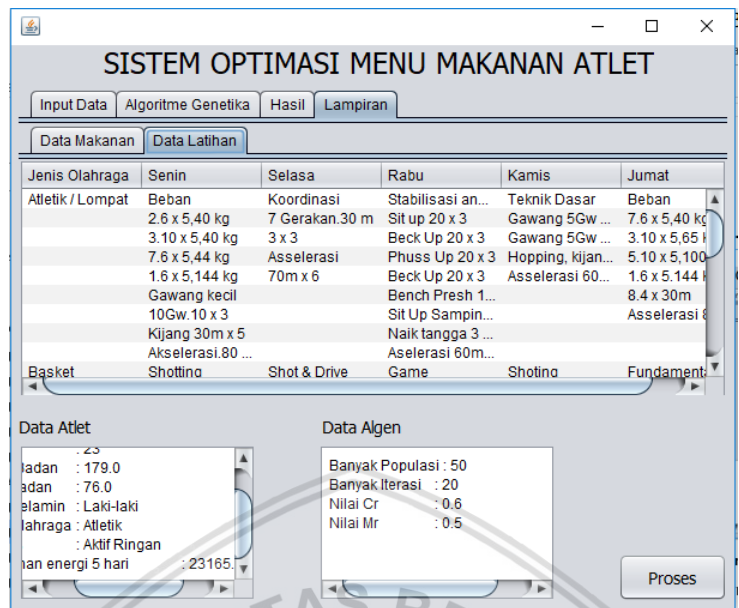
5.3.4 Implementasi antarmuka halaman lampiran

Halaman lampiran ini menampilkan seluruh daftar makanan dan daftar latihan yang digunakan pada sistem ini. Halaman lampiran ini dapat dilihat pada Gambar 5.12 dan Gambar 5.13., dimana implementasi antar muka ini mengacu pada perancangan Gambar 4.16 dan Gambar 4.17.



Gambar 0.12 Antarmuka Halaman Lampiran Data Makanan





Gambar 0.13 Antarmuka Halaman Lampiran Data Latihan



BAB 6 PENGUJIAN DAN PEMBAHASAN

Bab ini akan menjelaskan tentang pengujian yang dilakukan dan hasil yang didapatkan berdasarkan skenario pengujian yang digunakan. Pengujian tersebut terdiri dari pengujian pengujian kombinasi nilai cr , mr , pengujian ukuran populasi, dan ukuran generasi.

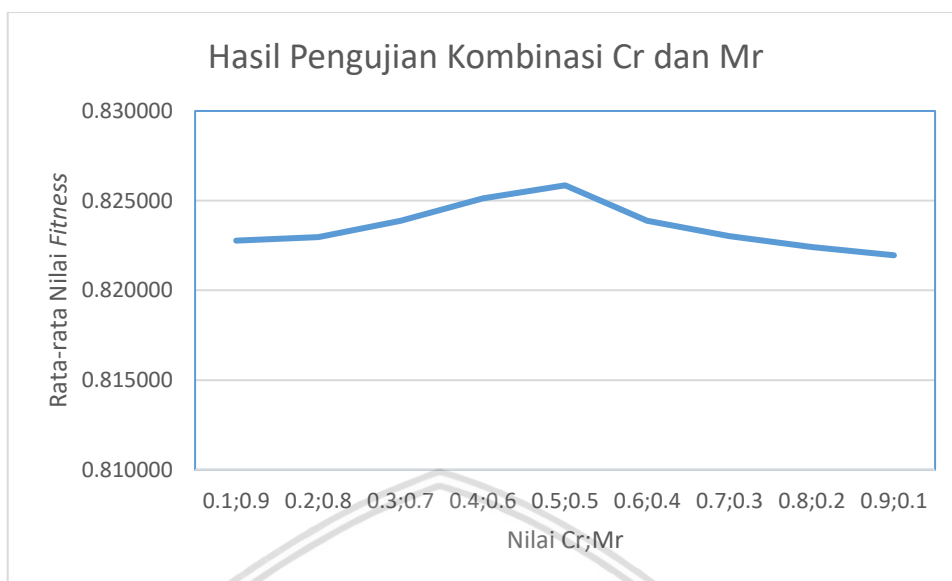
6.1 Pengujian kombinasi *crossover rate* dan *mutation rate*

Pada pengujian kombinasi *crossover rate* (cr) dan *mutation rate* (mr) dilakukan uji coba dengan kombinasi nilai cr dan mr antara 0,1 hingga 0,9. Pengujian kombinasi *crossover rate* (cr) dan *mutation rate* (mr) mengacu pada perancangan pengujian pada Tabel 4.20. Pengujian pertama kali dilakukan dengan menggunakan ukuran populasi sebesar 100 dengan ukuran generasi sebesar 1000, dimana dalam pengujian ini mendapatkan hasil yang sama ketika telah memasuki generasi ke 450. Selanjutnya karena telah mendapatkan nilai yang sama, pengujian selanjutnya dilakukan sebanyak 10 kali dengan menggunakan ukuran populasi sebesar 100 dan ukuran generasi sebesar 500. Hasil Pengujian kombinasi cr dan mr dapat dilihat pada Tabel 6.1.

Tabel 0.1 Hasil Pengujian Kombinasi *Crossover Rate* dan *Mutation Rate*

Cr;Mr	Pengujian ke										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0,1;0,9	0,82374	0,82328	0,81848	0,82654	0,82184	0,82374	0,82328	0,81848	0,82654	0,82184	0,822776
0,2;0,8	0,82006	0,82272	0,82497	0,82683	0,82023	0,82006	0,82272	0,82497	0,82683	0,82023	0,822963
0,3;0,7	0,82227	0,82232	0,82653	0,82511	0,82317	0,82227	0,82232	0,82653	0,82511	0,82317	0,823879
0,4;0,6	0,82957	0,82286	0,82608	0,82553	0,82162	0,82957	0,82286	0,82608	0,82553	0,82162	0,825133
0,5;0,5	0,82458	0,82788	0,82686	0,82537	0,82458	0,82458	0,82788	0,82686	0,82537	0,82458	0,825855
0,6;0,4	0,82429	0,82442	0,82398	0,82109	0,82563	0,82429	0,82442	0,82398	0,82109	0,82563	0,823884
0,7;0,3	0,82059	0,82494	0,82059	0,82510	0,82385	0,82059	0,82494	0,82059	0,82510	0,82385	0,823016
0,8;0,2	0,82328	0,82101	0,82302	0,82414	0,82062	0,82328	0,82101	0,82302	0,82414	0,82062	0,822415
0,9;0,1	0,82119	0,82509	0,82019	0,82177	0,82155	0,82119	0,82509	0,82019	0,82177	0,82155	0,821957

Berdasarkan hasil pengujian pada Tabel 6.1 dapat dilihat grafik hasil pengujian seperti ditunjukkan pada Gambar 6.1.



Gambar 0.1 Grafik Hasil Pengujian Kombinasi *Crossover Rate* dan *Mutation Rate*

Pada Tabel 6.1 dapat dilihat rata-rata nilai *fitness* tertinggi didapatkan pada saat kombinasi nilai *cr* 0,5 dan *mr* 0,5 yaitu sebesar 0,825855 dan rata-rata nilai *fitness* terendah ketika kombinasi nilai *cr* 0,9 dan *mr* 0,1 yaitu sebesar 0,821957. Gambar 6.1 menunjukkan grafik peningkatan rata-rata nilai *fitness* pada kombinasi nilai *cr* 0,1 dan *mr* 0,9 sampai dengan kombinasi nilai *cr* 0,5 dan *mr* 0,5. Hasil tersebut menunjukkan nilai *crossover rate* yang terlalu rendah dan *mutation rate* yang terlalu tinggi menyebabkan Algoritme genetika tidak dapat memperluas area pencarian, dan sebaliknya nilai *crossover rate* yang terlalu tinggi dan *mutation rate* yang terlalu rendah menyebabkan algoritme genetika tidak dapat melakukan eksplorasi ruang pencarian yang lebih efektif.

6.2 Pengujian ukuran populasi

Pada pengujian ukuran populasi dilakukan uji coba terhadap ukuran populasi kelipatan 10 dengan masing-masing 10 kali uji coba. Dalam pengujian ukuran populasi ini mengacu pada perancangan pengujian Tabel 4.21. Dimana ukuran populasi yang diuji yaitu ukuran populasi 10 hingga 100. Pengujian ukuran populasi ini menggunakan kombinasi nilai *cr* dan *mr* adalah 0,5 dan 0,5 dengan ukuran generasi sebesar 500. Hasil Pengujian ukuran populasi dapat dilihat pada Tabel 6.2.

Tabel 0.2 Hasil Pengujian Ukuran Populasi

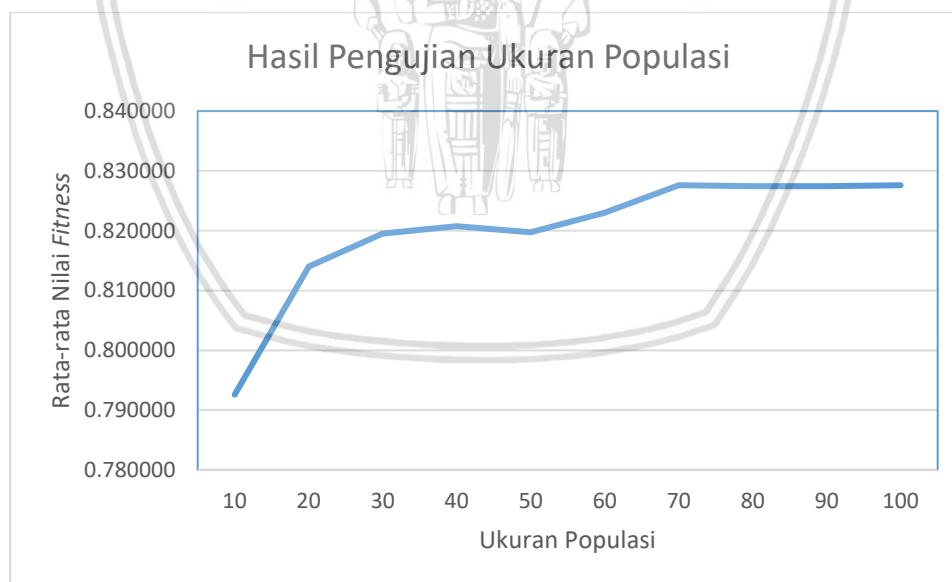
Populasi	Pengujian ke										Rata-rata Nilai Fitness
	1	2	3	4	5	6	7	8	9	10	
10	0,79663	0,79928	0,81441	0,77663	0,77589	0,79663	0,79928	0,81441	0,77663	0,77589	0,792569
20	0,82090	0,80849	0,81816	0,80977	0,81259	0,82090	0,80849	0,81816	0,80977	0,81259	0,813983
30	0,81747	0,81833	0,82377	0,81752	0,82057	0,81747	0,81833	0,82377	0,81752	0,82057	0,819534
40	0,81710	0,82178	0,82128	0,81982	0,82383	0,81710	0,82178	0,82128	0,81982	0,82383	0,820761

Tabel 6.2 Hasil Pengujian Ukuran Populasi (Lanjut)

Populasi	Pengujian ke										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50	0,82041	0,82195	0,82260	0,81651	0,81726	0,82041	0,82195	0,82260	0,81651	0,81726	0,819746
60	0,82775	0,81905	0,82082	0,81880	0,82856	0,82775	0,81905	0,82082	0,81880	0,82856	0,822996
70	0,82856	0,82856	0,82856	0,82375	0,82856	0,82856	0,82856	0,82856	0,82856	0,82375	0,827600
80	0,82856	0,82246	0,82856	0,82856	0,82856	0,82856	0,82856	0,82546	0,82654	0,82856	0,827440
90	0,82856	0,82546	0,82856	0,82856	0,82856	0,82246	0,82856	0,82856	0,82654	0,82856	0,827440
100	0,82856	0,82856	0,82856	0,82375	0,82856	0,82856	0,82856	0,82856	0,82375	0,82856	0,827600

Pada Tabel 6.2 hasil pengujian ukuran populasi dapat dilihat rata-rata nilai *fitness* tertinggi yang dihasilkan adalah 0,827600 yaitu pada ukuran populasi 70 dan rata-rata nilai *fitness* terendah adalah 0,792569 yaitu pada ukuran populasi 10, hal ini disebabkan ukuran populasi yang besar akan menghasilkan individu yang semakin beragam melalui proses *crossover* dan mutasi, dan berdampak pada nilai variasi yang dihasilkan, namun ukuran populasi yang besar juga tidak menjamin akan menghasilkan nilai *fitness* yang tinggi dikarenakan pembangkitan populasi yang dilakukan secara acak (Kusumaningsih, 2016). Ukuran populasi yang terlalu kecil akan memperkecil peluang untuk memperoleh solusi terbaik.

Berdasarkan hasil pengujian ukuran populasi pada Tabel 6.2 dapat dilihat grafik hasil pengujian seperti ditunjukkan pada Gambar 6.2.



Gambar 0.2 Grafik Hasil Pengujian Ukuran Populasi

Pada Gambar 6.2 dapat dilihat rata-rata nilai *fitness* mengalami kenaikan pada ukuran populasi 10 hingga 70. Pada ukuran populasi 70 rata-rata nilai *fitness* tidak menunjukkan perubahan yang signifikan pada ukuran generasi 80 dan 90, sedangkan memiliki nilai yang sama pada ukuran generasi 100 sehingga ukuran populasi 70 menjadi ukuran populasi yang optimal.

6.3 Pengujian ukuran generasi

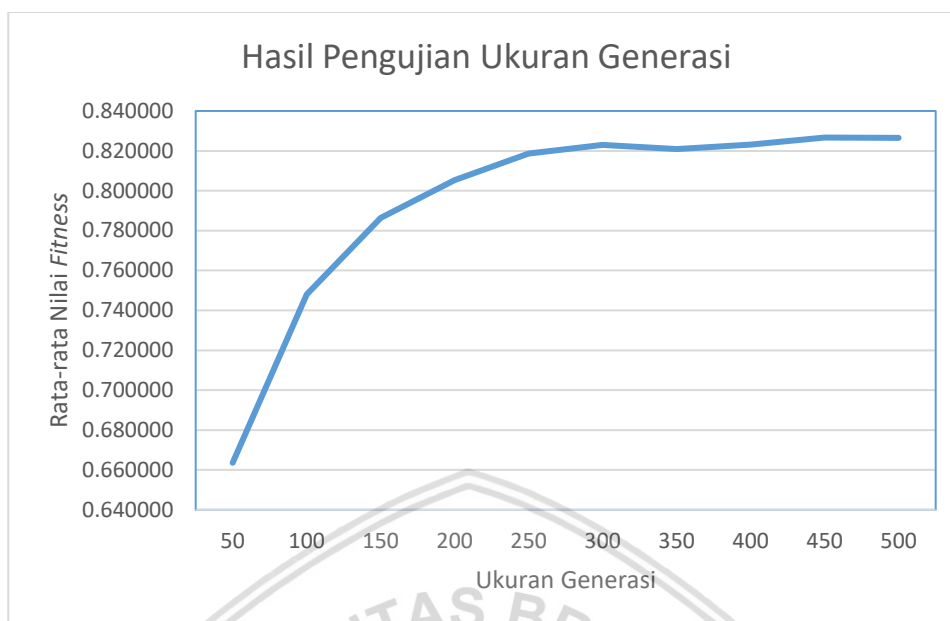
Pengujian ukuran generasi ini dilakukan berdasarkan perancangan pengujian ukuran generasi seperti pada Tabel 4.22. Pengujian ukuran generasi yang dilakukan yaitu ukuran generasi 50 hingga 500. Pengujian ini dilakukan untuk mengetahui bagaimana pengaruh jumlah generasi terhadap rata-rata nilai *fitness* yang didapatkan. Ukuran generasi yang diuji adalah 50, 100, 150, 200, 250, 300, 350, 400, 450, dan 500. Setiap pengujian ukuran generasi dilakukan sebanyak 10 kali uji coba. Ukuran populasi yang digunakan yaitu 100 dengan kombinasi nilai *cr* dan *mr* adalah 0.5 dan 0.5. Hasil pengujian jumlah generasi ditunjukkan pada Tabel 6.3.

Tabel 0.3 Hasil Pengujian Ukuran Generasi

Generasi	Pengujian ke										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50	0,65919	0,66640	0,65919	0,67846	0,65782	0,66640	0,65919	0,67232	0,65919	0,65782	0,663600
100	0,71647	0,74872	0,75939	0,75662	0,74841	0,74872	0,74872	0,75662	0,74841	0,74841	0,748049
150	0,77966	0,79423	0,78647	0,77467	0,78949	0,79423	0,78949	0,78647	0,77966	0,78949	0,786388
200	0,80429	0,80710	0,80238	0,80947	0,80154	0,80238	0,80429	0,80238	0,80947	0,80947	0,805274
250	0,82185	0,81696	0,82185	0,81647	0,82089	0,81647	0,81696	0,81696	0,81647	0,82185	0,818674
300	0,82380	0,82303	0,82353	0,82146	0,82330	0,82380	0,82303	0,82353	0,82146	0,82330	0,823026
350	0,81880	0,82284	0,81948	0,82319	0,82030	0,81880	0,82284	0,81948	0,82319	0,82030	0,820920
400	0,82569	0,81937	0,82301	0,82585	0,82199	0,82569	0,81937	0,82301	0,82585	0,82199	0,823180
450	0,82667	0,82667	0,82667	0,82667	0,82667	0,82667	0,82667	0,82667	0,82667	0,82667	0,826675
500	0,82667	0,82667	0,82667	0,82667	0,82575	0,82667	0,82667	0,82667	0,82667	0,82575	0,826491

Pada Tabel 6.3 hasil pengujian ukuran generasi dapat dilihat rata-rata nilai *fitness* tertinggi yang dihasilkan adalah 0,826675 yaitu pada ukuran generasi 450, rata-rata nilai *fitness* terendah adalah 0,663600 yaitu pada ukuran generasi 50 hal ini disebabkan ukuran generasi yang kecil membatasi eksplorasi ruang pencarian sebaliknya ukuran generasi yang besar memberi peluang untuk melakukan eksplorasi ruang pencarian yang lebih besar.

Berdasarkan hasil pengujian pada Tabel 6.3 dapat dilihat pengaruh ukuran generasi terhadap rata-rata nilai *fitness* yang didapatkan seperti ditunjukkan pada Gambar 6.3.



Gambar 0.3 Grafik Hasil Pengujian Ukuran Generasi

Pada Gambar 6.3 menunjukkan rata-rata nilai *fitness* mengalami kenaikan pada generasi 50 hingga generasi 450 dan setelah generasi 450 rata-rata nilai *fitness* tidak mengalami perubahan. Apabila pengujian ukuran generasi dilanjutkan dengan ukuran generasi lebih tinggi maka nilai *fitness* tidak memiliki selisih yang besar dan individu yang dihasilkan tidak berbeda jauh dengan induknya (Shafaat, et al., 2018). Berdasarkan hasil tersebut maka ukuran generasi 450 dapat dianggap optimal untuk digunakan dalam optimasi menu makanan atlet berdasarkan jadwal latihan.

6.4 Hasil analisis global

Berdasarkan hasil pengujian ukuran generasi, ukuran populasi dan kombinasi *crossover rate* dan *mutation rate* didapatkan nilai optimal parameter algoritme genetika sebagai berikut:

- Ukuran generasi : 450
- Ukuran populasi : 70
- *Crossover rate* : 0,5
- *Mutation rate* : 0,5

Parameter-parameter hasil pengujian tersebut selanjutnya digunakan dalam pengujian terhadap data atlet sebagai berikut:

- Nama : MY
- Usia : 21 tahun
- Tinggi badan : 172 cm

- Berat badan : 65 kg
- Aktivitas : Aktif Ringan
- Cabang Olahraga : Basket

Berdasarkan data atlet di atas didapatkan hasil perhitungan kebutuhan gizi harian sebagai berikut:

- Kalori : 1938 kkal
- Karbohidrat : 354,925 gram
- Protein : 92,675 gram
- Lemak : 53,067 gram

Berdasarkan parameter algoritme dan parameter data atlet yang diuji didapatkan hasil rekomendasi sistem dengan kandungan gizi seperti ditunjukkan pada Tabel 6.4.

Tabel 0.4 Kandungan Gizi Sistem

Hari ke-	Kalori	Karbohidrat	Protein	Lemak	Harga
1	4421	515,2	147,7	190,2	36500
2	4489	474,8	157,3	209,9	40800
3	4004,9	579,7	141,7	136,8	32400
4	4170	561,9	149,8	135,9	39500
5	4624,2	505,7	163,8	206,5	44400

Pada Tabel 6.4 dapat dilihat hasil rekomendasi sistem memiliki kandungan gizi yang kurang sesuai dengan kebutuhan gizi aktual atlet. Selisih kebutuhan gizi aktual dengan kebutuhan gizi hasil rekomendasi sistem dapat dilihat pada Tabel 6.5.

Tabel 0.5 Perbandingan Kebutuhan Gizi Aktual dan Sistem

Hari ke-	Kalori	Karbohidrat	Protein	Lemak
1	183	3,955	10,82	12,45
	(4,14%)	(0,77%)	(7,33%)	(6,55%)
2	401	25,98	6,22	14,80
	(8,93%)	(5,47%)	(3,95%)	(7,05%)
3	333,1	48,295	11,46	11,76
	(8,32%)	(8,33%)	(8,09%)	(8,60%)
4	32	54,995	8,21	9,21
	(0,77%)	(9,79%)	(5,48%)	(6,78%)
5	336,2	19,58	7,72	13,87
	(7,27%)	(3,87%)	(4,71%)	(6,72%)
Rata-rata selisih	257,06	30,561	8,886	12,418
	(5,88%)	(5,64%)	(5,91%)	(7,13%)



Pada Tabel 6.5 dapat dilihat rata-rata selisih kandungan gizi makanan hasil rekomendasi sistem dengan rata-rata selisih kalori sebesar 5,88%, rata-rata selisih karbohidrat 5,64%, rata-rata selisih protein sebesar 5,91%, dan selisih lemak sebesar 7,13%. Adanya selisih antara kebutuhan aktual atlet dan rekomendasi sistem disebabkan karena proses inialisasi populasi awal yang secara acak sehingga kemungkinan individu yang dihasilkan masih jauh dari hasil solusi optimal, selain itu data yang digunakan masih terbatas sehingga eksplorasi pencarian solusi menjadi terbatas.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai optimasi menu makanan atlet berdasarkan jadwal latihan didapatkan kesimpulan sebagai berikut

1. Dalam implementasi sistem optimasi menu makanan atlet menggunakan algoritme genetika, terdapat beberapa parameter yang mempengaruhi proses optimasi, seperti kombinasi nilai cr dan mr , ukuran populasi, dan banyak generasi. Pada pengujian kombinasi nilai cr dan mr yang telah dilakukan, didapatkan nilai kombinasi cr 0,5 dan mr 0,5 adalah kombinasi paling optimal dimana nilai *fitness* yang dihasilkan sebesar 0,825855 sedangkan kombinasi terendah yaitu kombinasi nilai cr 0,9 dan mr 0,1 dengan nilai *fitness* sebesar 0,821957. Pada pengujian ukuran populasi didapatkan nilai *fitness* tertinggi sudah didapatkan pada ukuran populasi 70 dengan nilai *fitness* sebesar 0,827600 dan rata-rata nilai *fitness* terendah adalah 0,792569 yaitu pada ukuran populasi 10. Dimana ukuran populasi yang besar akan menghasilkan individu yang semakin beragam melalui proses *crossover* dan mutasi, dan berdampak pada nilai variasi yang dihasilkan. Selanjutnya pada pengujian ukuran generasi dimana pengujian yang dilakukan menggunakan ukuran generasi dari 50 sampai 500, dan didapatkan ukuran generasi sudah mencapai nilai optimal pada ukuran 450 dengan nilai *fitness* sebesar 0,826675. Hal ini menunjukkan semakin besarnya ukuran generasi maka semakin besar ruang eksplorasi yang dimiliki, namun ukuran generasi yang telah mencapai nilai optimal jika dilanjutkan dengan nilai yang lebih besar maka nilai *fitness* dan individu yang dihasilkan tidak akan berbeda jauh dengan hasil sebelumnya.
2. Kualitas hasil rekomendasi makanan sistem dapat diukur dari hasil rata-rata selisih gizi makanan hasil rekomendasi sistem kebutuhan gizi aktual, dimana didapatkan rata-rata selisih kalori sebesar 5,88%, rata-rata selisih karbohidrat 5,64%, rata-rata selisih protein sebesar 5,91%, dan selisih lemak sebesar 7,13%.

7.2 Saran

Untuk pengembangan pada penelitian selanjutnya terdapat beberapa saran yang diharapkan dapat memperbaiki hasil penelitian yaitu sebagai berikut:

1. Data kebutuhan energi latihan atlet yang digunakan lebih fleksibel dan dapat diupdate guna memberikan nilai kebutuhan gizi aktual atlet lebih baik lagi.
2. Pada penelitian selanjutnya diharapkan dapat memodifikasi perhitungan nilai *fitness* yang digunakan agar dapat memberikan hasil yang lebih baik dan solusi yang diberikan sesuai dengan kebutuhan gizi.

3. Pada penelitian selanjutnya diharapkan dapat melakukan modifikasi representasi kromosom yang dimana dapat melakukan perhitungan langsung untuk banyak atlet. Selain itu perhitungan energi harian dapat dilakukan perhari, karena perhitungan kebutuhan energi harian menggunakan nilai aktivitas fisik yang memungkinkan setiap hari aktivitas fisik yang dilakukan berbeda-beda.



DAFTAR PUSTAKA

- Anonim, 2016. *jpnn.com*. [Online]
Available at: <https://www.jpnn.com/news/makanan-kurang-atlet-teriak-piring-pun-dibalikkan>
[Diakses 09 09 2017].
- Gen, M. & Cheng, R., 1997. *Genetic Algorithm & Engineering Design*. Ashikaga: s.n.
- Haupt, R. L. & Haupt, S. E., 2004. *Practical Genetic Algorithm*. New Jersey: John Wiley & Sons, Inc.
- Kemenkes, 2014. *Pedoman Gizi Olahraga Prestasi*. s.l.:s.n.
- Kusumaningsih, F. D., 2016. *Penerapan Algoritma Genetika Pada Optimasi Susunan Bahan Makanan Untuk Pemenuhan Kebutuhan Gizi Keluarga*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.
- Li, Y. & Chen, Y., 2010. *A Genetic Algorithm for Job-Shop Scheduling*. China: s.n.
- Mahmudy, W. F., 2013. *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
- Mahmudy, W. F., Marian, R. M. & Luong, L. H. S., 2013. *Optimization of part type selection and loading problem with alternative production plans in flexible manufacturing system using hybrid genetic algorithms*. Chonburi: s.n.
- Maryamah, Putri, R. R. M. & Wicaksono, S. A., 2017. Optimasi Komposisi Makanan Pada Penderita Diabetes Melitus dan Komplikasinya Menggunakan Algoritma Genetika. *Jurnal Pengembangan Teknologi Inormasi dan Ilmu Komputer*, pp. 270-281.
- Safrizal, 2014. *GoRiau.com*. [Online]
Available at: <https://www.goriau.com/berita/kep-meranti/atlit-panjat-tebing-asal-meranti-gadai-hape-untuk-biaya-makan-di-pekanbaru.html>
[Diakses 09 09 2017].
- Shafaat, M., Cholissodin, I. & Santoso, E., 2018. Optimasi Komposisi Makanan Diet Bagi Penderita Hipertensi menggunakan Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 226-236.
- Sihombing, R. S., 2014. *Pemanfaatan Algoritma Genetika pada Aplikasi Penempatan Buku untuk Perpustakaan Sekolah*. Medan: STMIK Budidarma.
- Wahid, N. & Mahmudy, W. F., 2015. Optimasi Komposisi Makanan Untuk Penderita Kolesterol Menggunakan Algoritma Genetika. *DORO:Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*.