

**OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET
SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA
UNTUK MEMENUHI KECUKUPAN GIZI**

SKRIPSI

Oleh:
MUHAMMAD IDHARUL KAMIL
NIM. 07650027



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET
SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA
UNTUK MEMENUHI KECUKUPAN GIZI**

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
MUHAMMAD IDHARUL KAMIL
NIM. 07650027

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET
SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA**

UNTUK MEMENUHI KECUKUPAN GIZI

SKRIPSI

Oleh:

MUHAMMAD IDHARUL KAMIL

NIM. 07650027

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 07 Juli 2014

Dosen Pembimbing I

Dosen pembimbing II

Ririen Kusumawati, M. Kom
NIP. 197203092005012002

Dr. Muhammad Faisal, M. T
NIP. 197712012008011007

Mengetahui
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian, M.CS
NIP. 19740424 200901 1 008

**OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET
SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA
UNTUK MEMENUHI KECUKUPAN GIZI**

SKRIPSI

Oleh:
MUHAMMAD IDHARUL KAMIL
NIM. 07650027

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal: 14 Juli 2014

Susunan Dewan Penguji	Tanda Tangan
1. Penguji Utama : <u>Dr. Cahyo Crysdian, M.CS</u> NIP. 197404242009011008	()
2. Ketua : <u>Fatchurrohman, M.Kom</u> NIP. 197007312005011002	()
3. Sekretaris : <u>Ririen Kusumawati, M. Kom</u> NIP. 197203092005012002	()
4. Anggota : <u>Dr. Muhammad Faisal, M. T</u> NIP. 197405102005011007	()

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian, M.CS
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini :

Nama : Muhammad Idharul Kamil
NIM : 07650027
Fakultas / Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : **OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA UNTUK MEMENUHI KECUKUPAN GIZI**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 2014

Yang Menyatakan,

Muhammad Idharul Kamil

NIM. 07650027

Motto

بِسْمِ الرَّحْمَنِ الرَّحِيمِ

وَإِذْ تَأَذَّنَ رَبُّكُمْ لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ وَلَئِن كَفَرْتُمْ إِنَّ عَذَابِي لَشَدِيدٌ

dan (ingatlah juga), tatkala Tuhanmu memaklumkan; "Sesungguhnya jika kamu bersyukur, pasti Kami akan menambah (nikmat) kepadamu, dan jika kamu mengingkari (nikmat-Ku), Maka Sesungguhnya azab-Ku sangat pedih".

LEMBAR PERSEMBAHAN

Yang Utama Dari Segalanya...

Sujud serta syukur kepada Allah SWT. Rohman Rohim Mu yang membuat hamba ini kuat. Tiada daya dan upaya melainkan kemurahanMu Ya Robbi. Atas karunia serta kemudahan yang Engkau berikan akhirnya skripsi yang sederhana ini dapat terselesaikan. Sholawat dan salam selalu terlimpahkan kepada Nabi junjungan umat Rasulullah Muhammad SAW.

Tak lain dan tak bukan kupersembahkan karya sederhana ini kepada orang-orang yang sangat kukasihi dan kusayangi.

Keluarga Tercinta

Sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga kupersembahkan karya kecil ini kepada Bapak Mudoyo dan Ibu Asfiah yang selalu membuatku termotivasi dan selalu menyirami kasih sayang, selalu mendoakanku, selalu menasehatiku menjadi lebih baik,. Yang telah memberikan kasih sayang, segala dukungan, dan cinta kasih yang tiada terhingga yang tiada mungkin dapat kubalas hanya dengan selembar kertas yang bertuliskan kata cinta dan persembahan. Terima kasih Abiii,, Umiii,, hanya kesabaranmu menguatkanku. Dan tak lupa kepada kakakku Idham Muhtadin yang senantiasa memberikan semangat serta do'anya dengan setulus hati.

Terima Kasih

Terima kasih

Terima kasih...

KATA PENGANTAR

Puji syukur penulis sampaikan kehadiran Allah SWT karena atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“OPTIMASI PENENTUAN BAHAN PANGAN HARIAN ATLET SEPAKBOLA MENGGUNAKAN ALGORITMA GENETIKA UNTUK MEMENUHI KECUKUPAN GIZI”**

Skripsi ini merupakan tugas akhir yang diajukan untuk memenuhi syarat dalam memperoleh gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa adanya bantuan dan dorongan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Prof. Dr. H. Mudjia Rahardjo M.Sc, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Ibu Dr. Hj. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Saintek Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Bapak Dr. Cahyo Crys dian M.CS selaku Ketua Jurusan Teknologi Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Ibu Ririen Kusumawati M. Kom selaku Dosen Pembimbing I sekaligus dosen wali atas bimbingan, pengarahan, dan kesabarannya hingga penulisan skripsi ini dapat terselesaikan.

5. Bapak Dr. Muhammad Faisal, M. T selaku Dosen Pembimbing II atas bimbingan, pengarahan, dan kesabarannya dalam membimbing penulisan skripsi ini yang terkait dengan agama hingga penulisan skripsi ini dapat terselesaikan.
6. Seluruh Dosen Teknik Informatika dan segenap perangkat Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
7. Bapak, Ibu dan Saudara-saudaraku yang tidak hentinya memberikan dukungan dan doa untuk melaksanakan skripsi dan mendapatkan hasil yang terbaik.
8. Lidya Deviga yang selalu memberikan perhatiannya dari segi waktu, tenaga, dan biaya saat penelitian, juga menemani ku menjalani perkuliahan serta dukungan dan semangat dalam skripsi ini.
9. Teman-teman Teknik Informatika khususnya angkatan 2007 atas segala kebersamaannya dari awal masuk hingga lulus kuliah.
10. Teman seperjuanganku Andi, diyu dan teman-teman angkatan TI 07. Tak lupa semua pihak yang telah memotivasi dan membantu dalam proses penyelesaian skripsi ini yang tidak bisa disebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan. Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan khususnya bermanfaat bagi penulis secara pribadi.

Malang, 05 Juli 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGAJUAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN.....	v
MOTTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
ABSTRAK.....	xiv
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat Penelitian.....	4
1.5 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	
2.1 Zat Gizi.....	6
2.1.1 Kecukupan Gizi.....	9
2.1.2 Perhitungan Kebutuhan Gizi Remaja.....	13
2.2 Bahan Pangan.....	16
2.3 Optimasi.....	16
2.4 Algoritma Genetika.....	17
2.4.1 Pengertian.....	17
2.4.2 Struktur Algoritma Genetika.....	22
2.4.3 Komponen Utama Algoritma Genetika.....	23
2.4.4 Nilai Fitnes.....	28
2.4.5 Seleksi.....	29
2.4.6 <i>Crossover</i>	34
2.4.7 Mutasi.....	35
BAB III ANALISIS DAN PERANCANGAN SISTEM	
3.1 Analisis Sistem.....	38
3.1.1 Pembangkitan Populasi Awal.....	38
3.1.2 Fungsi Fitnes.....	43
3.1.3 Proses Seleksi.....	44
3.1.4 Proses Kawin Silang.....	45
3.1.5 Proses Mutasi.....	46

3.2 Perancangan Sistem.....	47
3.2.1 Perancangan Diagram Alir.....	48
3.2.2 Data Flow Diagram.....	49
3.2.3 Entit Relationship Diagram.....	50
3.2.4 Struktur Tabel	52
BAB IV IMPLEMENTASI DAN PEMBAHASAN	
4.1 Implementasi	56
4.2 Penjelasan Program	56
4.2.1 Form Awal.....	56
4.2.2 Form Data Bahan Pangan.....	58
4.2.3 Form Atlet	58
4.2.4 Form Optimasi.....	60
4.3 Deskripsi Sistem.....	61
4.3.1 Inisialisasi Populasi	61
4.3.2 Hitung Fitnes	62
4.3.3 Seleksi	63
4.3.4 <i>Crossover</i>	65
4.3.5 Mutasi	66
4.3.6 Hasil Akhir	67
4.3.7 Simpan Hasil	68
4.4 Uji Coba Aplikasi	69
4.5 Optimasi Dalam Pandangan Islam	78
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan.....	80
5.2 Saran.....	81
DAFTAR PUSTAKA	

DAFTAR TABEL

Tabel 2.1	Tabel (BMR) Laki-laki Berdasarkan Berat Badan	10
Tabel 2.2	Tabel (BMR) Perempuan Berdasarkan Berat Badan	11
Tabel 2.3	Tabel Rata-rata Tingkat Aktivitas Harian	12
Tabel 2.4	Tabel Kebutuhan Energi Aktivitas Olahraga Berdasar Berat Badan	12
Tabel 2.5	Tabel Kebutuhan Energi Untuk Pertumbuhan	13
Tabel 2.6	Tabel Ukuran Status Gizi Menurut IMT	14
Tabel 2.7	Tabel Golongan Bahan Pangan	16
Tabel 3.1	Tabel Kelompok 1 PADI-padian dan Hasil Olahanya	39
Tabel 3.2	Tabel Kelompok 2 Umbi-umbian dan Hasil Olahanya	39
Tabel 3.3	Tabel Kelompok 3 Pangan Hewani dan Hasil Olahanya	40
Tabel 3.4	Tabel Kelompok 4 Kacang-kacangan dan Hasil Olahanya	40
Tabel 3.5	Tabel Kelompok 5 Buah-biji Berminyak dan Hasil Olahanya	41
Tabel 3.6	Tabel Kelompok 6 Minyak, Lemak dan Hasil Olahanya	41
Tabel 3.7	Tabel Kelompok 7 Gula dan Hasil Olahanya	41
Tabel 3.8	Tabel Kelompok 8 Sayur dan Buah	42
Tabel 3.9	Tabel Kelompok 9 Lain-lain	42
Tabel 3.10	Tabel Makanan	53
Tabel 3.11	Tabel Atlet	53
Tabel 3.12	Tabel Hasil	54
Tabel 3.13	Tabel Detail Hasil	55

DAFTAR GAMBAR

Gambar 2.1	Ilustrasi Istilah Pada Algoritma Genetika	21
Gambar 2.2	Siklus Sederhana Algoritma Genetika	22
Gambar 2.3	<i>Flowchart</i> Algoritma Genetika	23
Gambar 2.4	Contoh Populasi 5 Kromosom	32
Gambar 2.5	Probabilitas Kromosom Dalam Roda <i>Roulette</i>	32
Gambar 2.6	<i>Crossover</i> 1 Titik	35
Gambar 2.7	<i>Crossover</i> 2 Titik	35
Gambar 2.8	Mutasi Tingkat Kromosom	36
Gambar 2.9	Mutasi Tingkat Gen	37
Gambar 2.10	Mutasi Tingkat Bit	37
Gambar 3.1	<i>Crossover</i> Dua Titik.....	45
Gambar 3.2	Input-Proses-Output	47
Gambar 3.3	Diagram Sederhana Algoritma Genetika	48
Gambar 3.4	DFD Level 1.....	50
Gambar 3.5	DFD Level 2.....	50
Gambar 3.6	CDM.....	51
Gambar 3.7	PDM	52
Gambar 4.1	Tampilan Form Pertama <i>Software</i>	57
Gambar 4.2	Tampilan Form Bahan Pangan.....	58
Gambar 4.3	Tampilan Form Atlet.....	59
Gambar 4.4	Tampilan Form optimasi.....	60
Gambar 4.5	Tampilan Input Data Atlet.....	69
Gambar 4.6	Tampilan Proses Genetika	73
Gambar 4.7	Tampilan Proses <i>Crossover</i>	74
Gambar 4.8	Tampilan Proses Mutasi.....	76
Gambar 4.9	Tampilan Hasil Akhir	78

ABSTRAK

Idharul Kamil, Muhammad. 2014. **Optimasi Penentuan Bahan Pangan Harian Atlet Sepakbola Menggunakan Algoritma Genetika Untuk Memenuhi Kebutuhan Gizi**. Skripsi, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Dosen Pembimbing: Ririen Kusumawati M.Kom, dan Dr. Muhammad Faisal, M. T.

Kata Kunci: Algoritma Genetika, Optimisasi Komposisi Bahan Pangan.

Ada banyak kemungkinan kombinasi makanan yang dapat digunakan untuk menentukan bahan pangan atlet sepakbola agar memenuhi kecukupan gizi. Namun demikian, tidak semua kombinasi makanan akan memberikan solusi terbaik. Agar solusi terbaik dapat dicapai, penelitian untuk menentukan bahan pangan harian atlet sepakbola perlu dilakukan. Untuk mempermudah proses penentuan komposisi tersebut, dilakukan pembuatan perangkat lunak yang diharapkan berguna untuk mencapai kombinasi bahan pangan atlet yang baik dengan memperhatikan kebutuhan gizi atlet.

Algoritma genetika yang memiliki kehandalan dalam menghasilkan output yang optimal dapat dimanfaatkan untuk memecahkan masalah tersebut dengan bantuan piranti lunak. Pada piranti lunak yang dibuat, terdapat beberapa input yang dibutuhkan, yaitu jumlah kromosom awal, jumlah generasi, probabilitas crossover dan probabilitas mutasi. Hasil pemrosesan merupakan kombinasi makanan yang merepresentasikan penyelesaian masalah ini. Selain itu juga terdapat input data dari atlet untuk penghitungan kebutuhan protein dan kalori yang diperlukan atlet. Hanya kromosom terbaik yang akan diberikan sebagai hasil.

Berdasarkan penelitian yang telah dilakukan, digunakan 53 jenis makanan yang akan digunakan pada pengujian. Dari jenis makanan tersebut, dipilih 9 jenis makanan mengisi setiap kromosomna. Hasil pengujian berbeda tergantung parameter yang dimasukkan. Selain parameter genetika hasil kombinasi juga tergantung pada data dan aktifitas masing-masing atlet. Seperti dengan data dan kegiatan atlet Ahmad diketahui kebutuhan kalorinya 4106,22 dan protein 102,655. Dilanjutkan proses genetika dengan parameter jumlah generasi 50, populasi 80, probabilitas crossover 0,45, probabilitas mutasi 0,01 dihasilkan komposisi bahan pangan sebagai berikut : Beras giling, Tepung sagu, Daging ayam, Kacang panjang, Apel, Yoghurt, Gula Pasir, Buncis, Minyak kelapa. Komposisi tersebut terdapat pada individu dan generasi 1 dengan perbandingan protein 103% dan kalori 100%.

ABSTRACT

idharul kamil, Muhammad 2014. **Optimization Determination of football athletes daily food is by using genetic algorithms to fulfill the nutritios.** Thesis, department of informatics, Faculty of Science and Technology, Islamic University of Maulana Malik Ibrahim Malang, Supervisor Ririen Kusumawati M.kom and Dr. Muhammad Faisal,M.T

Keywords: *Genetic Algorithms, Optimization of Food Composition.*

There are many possible combinations of foods that can determine food for football athletes in order to meet their nutritional adequacy . However , not all combinations of foods will give best solutions. In order to achieve the best solution, a study to determine daily foods of football athletes need to be done. To simplify the process of determining composition, useful software is made which is expected to achieve a good combination of food athletes by focusing on nutritional needs of athletes.

Genetic algorithms that have reliability in producing optimum output can be used to solve the problems with software . In software, there are several inputs required, the number of initial chromosomes, number of generations , crossover probability and mutation probability. The result of processing is a combination of food that represents the completion of this problem. There are also data input from the athletes to the calculation of protein and calories needs of athletes required.Only best chromosome that will be awarded as a result.

Based on the research , used 53 types of food that will be used in testing. Of these foods, selected 9 kinds of food to fill every kromosomna. The result test vary depending on the parameters entered. In addition , the result combination of genetic parameters also depend on the data and activities of each athletes. Data and activities athletes ahmad 4106.22 calorie and 102.655 protein. Proess of genetic parameters 50 number of generations, population 80, crossover probability of 0.45,0.01 mutation probability produced food composition as follows : Milled rice, sago flour, chicken meat, longbeans, apples, yogurt, cane sugar, beans, coconut oil. The composition contained in the individual and generation 1 with ratio of 103% protein and 100% calories.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sepakbola sudah menjadi olahraga dunia yang masuk diberbagai kalangan. Itu disebabkan permainan yang bagus juga menghibur dan dapat dinikmati masyarakat. Ada beberapa faktor pendukung keberhasilan *team* sepakbola menyuguhkan permainan yang bagus, diantaranya adalah penampilan yang maksimal dari pemainnya.

Selain pengaturan latihan yang disiplin juga strategi jitu yang diinstruksikan oleh pelatih, faktor makanan yang dikonsumsi pemain tidak kalah pentingnya. Karena tubuh pemain memerlukan komposisi makanan yang tepat untuk menunjang permainan yang maksimal.

وَكُلُوا مِمَّا رَزَقَكُمُ اللَّهُ حَلَالًا طَيِّبًا وَاتَّقُوا اللَّهَ الَّذِي أَنْتُمْ بِهِ مُؤْمِنُونَ (Al-Maidah88)

88. dan makanlah makanan yang halal lagi baik dari apa yang Allah telah rezekikan kepadamu, dan bertakwalah kepada Allah yang kamu beriman kepada-Nya.

Tafsir Ibnu Katsir : Setelah Allah Ta'ala menjelaskan bahwa tidak ada tuhan melainkan Dia, maka Dia menjelaskan bahwa Dia Maha Pemberi rezeki kepada seluruh makhluk-Nya. Dia menganugerahkan kepada mereka kebolehan memakan makanan yang halal lagi baik serta melarang mereka memakan makanan yang diharamkan kepadanya.

Pada ayat di atas Allah SWT memerintahkan kepada manusia untuk memakan makanan yang halal. makanan halal itu adalah makanan yang didapat dan dimakan dengan cara yang diperbolehkan agama, yang tidak mengandung unsur haram di dalamnya. Selain itu Allah SWT juga memerintahkan untuk memakan makanan yang baik. Makanan yang baik yaitu makanan yang memenuhi kebutuhan gizi bagi tubuh manusia, dengan makanan itu tubuh menjadi sehat.

Perintah tersebut untuk seluruh manusia agar tidak ada makanan haram yang masuk dalam tubuh. Juga agar manusia sehat karena kebutuhan tubuh terpenuhi dan dapat beraktifitas dengan baik. Karena setiap aktifitas yang dikerjakan manusia memerlukan kebutuhan asupan yang berbeda, maka dari itu diperlukan komposisi pangan sesuai dengan aktifitas yang dikerjakan.

Komposisi makanan bagi atlet sepakbola pastilah berbeda dengan masyarakat umum. Karena aktifitas yang ekstra, komposisi pangan atlet harus diatur agar memenuhi kebutuhan tubuhnya. Kebutuhan tubuh yang tidak terpenuhi menyebabkan tubuh sakit, apalagi seorang atlet yang lebih banyak aktifitasnya.

Dengan menggunakan konsep optimasi, maka akan dihasilkan suatu keluaran berupa komposisi dari bahan pangan yang sebaiknya dikonsumsi untuk memenuhi kebutuhan nutrisi didalam kehidupan sehari-hari bagi atlet sepakbola. Algoritma genetika yang memiliki kehandalan dalam menghasilkan *output* yang optimal, dapat dimanfaatkan untuk kepentingan tersebut.

1.2 Rumusan Masalah

Bagaimana membangun *software* pendukung keputusan dengan menggunakan algoritma genetika untuk menentukan komposisi bahan pangan harian bagi atlet sepakbola.

1.3 Batasan Masalah

Batasan permasalahan dalam pembuatan aplikasi ini meliputi:

1. Aplikasi ini hanya menentukan komposisi bahan pangan dan takaran masing-masing per hari, tidak menentukan penyusunan menu makan.
2. Makanan yang disajikan adalah makanan yang halal.
3. Penentuan komposisi bahan pangan dan takarannya menggunakan algoritma genetika.
4. Gizi yang masuk dalam perhitungan penentuan takaran bahan pangan per hari adalah energi (kalori) dan protein.
5. Menggunakan bahasa pemrograman Java dengan editor Netbean.
6. Menggunakan database MySQL sebagai basis datanya.

1.4 Tujuan Dan Manfaat

1. Tujuan

Untuk membuat aplikasi pendukung keputusan dengan menggunakan algoritma genetika untuk menentukan komposisi bahan pangan bagi atlet sepakbola.

2. Manfaat Bagi Peneliti

Menambah wawasan keilmuan dan keterampilan dalam membangun aplikasi yang terdapat unsur kecerdasan buatan. Dan sebagai penguat keinginan peneliti untuk melanjutkan studi dibidang teknologi informasi.

3. Manfaat Bagi Umum

Penting aplikasi ini diterapkan agar atlet sehat dan kuat dalam latihan maupun pertandingan. Dengan aplikasi ini manajemen atlet dapat mengatur komposisi makanan bagi atletnya. Sehingga atlet tercukupi kebutuhan nutrisi dan maksimal permainannya. Dengan begitu club akan meraih banyak kemenangan.

1.5 Sistematika Penulisan

Dalam penulisan skripsi ini, secara keseluruhan terdiri dari lima bab yang masing-masing bab disusun dalam sistematika sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini menjelaskan tentang kecukupan dan kebutuhan gizi, teori dasar algoritma genetika, dan optimasi dengan algoritma genetika. Adapun literatur yang digunakan meliputi buku referensi dan dokumentasi *internet*.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan tahap analisis dan perancangan Optimasi Komposisi Bahan Pangan Harian Atlet Dengan Metode Algoritma Genetika.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini menjelaskan tahapan implementasi dan uji coba dari perancangan sistem.

BAB V : PENUTUP

Pada bab ini berisi kesimpulan dari pembahasan dan saran yang bermanfaat untuk pengembangan skripsi ini.

BAB II

TINJAUAN PUSTAKA

Pada bagian ini, akan dibahas tentang dasar ilmu dan teori-teori penunjang permasalahan skripsi ini.

2.1. Zat Gizi

Zat gizi merupakan unsur yang terkandung dalam makanan yang memberikan manfaat bagi kesehatan manusia. Masing-masing bahan makanan yang dikonsumsi memiliki kandungan gizi yang berbeda. Zat gizi yang terkandung dalam makanan tersebut berbeda-beda antara makanan yang satu dengan yang lainnya. Perbedaan tersebut dapat berupa jenis zat gizi yang terkandung dalam makanan, maupun jumlah dari masing-masing zat gizi.

Kebutuhan energi berbeda bagi setiap orang. Bervariasi tergantung umur, jenis kelamin dan aktifitas fisik. Anak - anak sangat membutuhkan nutrisi untuk perkembangannya sedang orang dewasa membutuhkannya untuk menjaga tubuh tetap sehat dan berkualitas.

Berikut ini perbandingan beberapa kebutuhan energi/kalori berdasarkan jenis kelamin, aktivitas fisik dan usia:

- anak-anak, wanita dewasa yang tidak banyak beraktivitas, dan dewasa tua membutuhkan sekitar 1600 kalori/hari
- remaja, wanita aktif dan laki-laki yang tidak beraktivitas membutuhkan sekitar 2000 kalori/hari

- remaja laki-laki yang aktif dan laki-laki dewasa muda membutuhkan sekitar 2400 kalori/hari. (Yusuf,Liswarti. 2008: 130)

Untuk memenuhi kebutuhan energi tersebut kita harus mengetahui zat-zat gizi apa saja yang terkandung dalam bahan makanan yang kita makan. Zat gizi bisa dikelompokkan menjadi beberapa hal yaitu berdasarkan fungsinya, berdasarkan jumlah yang dibutuhkan oleh tubuh dan berdasarkan sumbernya.

1) Berdasarkan fungsinya

Setiap zat gizi memiliki fungsi yang spesifik. Masing-masing zat gizi tidak dapat berdiri sendiri dalam membangun tubuh dan menjalankan proses metabolisme. Namun zat gizi tersebut memiliki berbagai fungsi yang berbeda.

a. Zat gizi sebagai sumber energi

Sebagai sumber energi zat gizi bermanfaat untuk menggerakkan tubuh dan proses metabolisme di dalam tubuh. Zat gizi yang tergolong kepada zat yang berfungsi memberikan energi adalah karbohidrat, lemak dan protein. Bahan pangan yang berfungsi sebagai sumber energi antara lain : nasi, jagung, talas merupakan sumber karbohidrat; margarine dan mentega merupakan sumber lemak; ikan, daging, telur dan sebagainya merupakan sumber protein.

- b. Zat gizi untuk pertumbuhan dan mempertahankan jaringan tubuh

Zat gizi ini memiliki fungsi sebagai pembentuk sel-sel pada jaringan tubuh manusia. Jika kekurangan mengkonsumsi zat gizi ini maka pertumbuhan dan perkembangan manusia akan terhambat. Selain itu zat gizi ini juga berfungsi untuk menggantikan sel-sel tubuh yang rusak dan mempertahankan fungsi organ tubuh.

Zat gizi yang termasuk dalam kelompok ini adalah protein, lemak, mineral dan vitamin. Namun zat gizi yang memiliki sumber dominan dalam proses pertumbuhan adalah protein.

- c. Zat gizi sebagai pengatur/ regulasi proses di dalam tubuh

Proses metabolisme di dalam tubuh perlu pengaturan agar terjadi keseimbangan. Untuk itu diperlukan sejumlah zat gizi untuk mengatur berlangsungnya metabolisme di dalam tubuh. Tubuh perlu keseimbangan, untuk itu proses metabolisme yang terjadi di dalam tubuh perlu di atur dengan baik. Zat gizi yang berfungsi untuk mengatur proses metabolisme di dalam tubuh adalah mineral, vitamin air dan protein. Namun yang memiliki fungsi utama sebagai zat pengatur adalah mineral dan vitamin.

2) Berdasarkan jumlah

Berdasarkan jumlah yang dibutuhkan oleh tubuh zat gizi terbagi atas dua, yaitu:

a. Zat gizi makro

Zat gizi makro adalah zat gizi yang dibutuhkan dalam jumlah besar dengan satuan gram. Zat gizi yang termasuk kelompok zat gizi makro adalah karbohidrat, lemak dan protein.

b. Zat gizi mikro

Zat gizi mikro adalah zat gizi yang dibutuhkan tubuh dalam jumlah kecil atau sedikit tapi ada dalam makanan. Zat gizi yang termasuk kelompok zat gizi mikro adalah mineral dan vitamin. Zat gizi mikro menggunakan satuan mg untuk sebagian besar mineral dan vitamin.

3) Berdasarkan Sumber

Berdasarkan sumbernya zat gizi terbagi dua, yaitu nabati dan hewani. (Almatsier,Sunita. 2003: 134)

2.1.1 Kecukupan Gizi

Sesuai prinsip dasar "Gizi Seimbang" yang mengandung cukup karbohidrat, lemak, protein, vitamin, mineral, air dan serat, maka kebutuhan gizi atlet sepakbola adalah sebagai berikut :

Energi

Secara umum seorang pemain sepakbola memerlukan energi sekitar 4.500 Kkal atau 1,5 kali kebutuhan energi orang dewasa normal dengan postur tubuh relatif sama, karena pemain sepakbola dikategorikan dengan seseorang yang melakukan aktivitas fisik yang berat.

Kebutuhan energi dihitung dengan memperhatikan beberapa komponen penggunaan energi yaitu : **Basal Metabolic Rate (BMR)**, **Specific Dynamic Action (SDA)**, **Aktivitas Fisik dan Faktor Pertumbuhan.**

a. Basal Metabolic Rate (BMR)

BMR merupakan jumlah energi yang dikeluarkan untuk aktivitas vital tubuh seperti denyut jantung, bernafas, transmisi elektrik pada otot dan lain-lain.

Tabel 2.1 : Tabel (BMR) laki-laki berdasarkan Berat Badan

Jenis Kelamin	Berat Badan (Kg)	Energi (Kal)		
		10-18 th	18-30 th	30-60 th
Laki-laki	55	1625	1514	1499
	60	1713	1589	1556
	65	1801	1664	1613
	70	1889	1739	1670
	75	1977	1814	1727
	80	2065	1889	1785
	85	2154	1964	1842
	90	2242	2039	1899

Tabel 2.2 : Tabel (BMR) perempuan berdasarkan Berat Badan

Jenis Kelamin	Berat Badan (Kg)	Energi (Kal)		
		10-18 th	18-30 th	30-60 th
Perempuan	40	1224	1075	1167
	45	1291	1149	1207
	50	1357	1223	1248
	55	1424	1296	1288
	60	1491	1370	1329
	65	1557	1444	1369
	70	1624	1518	1410
	75	1691	1592	1450

b. Specific Dynamic Action (SDA)

Merupakan jumlah *energy* yang dibutuhkan untuk mengolah makanan dalam tubuh, antara lain untuk proses pencernaan dan penyerapan zat-zat gizi oleh usus. Besarnya SDA kurang lebih 10% BMR.

c. Aktivitas Fisik

Pengeluaran energi untuk aktivitas fisik harian ditentukan oleh jenis, identitas dan lamanya aktivitas fisik dan olahraga.

Table 2.3 : Tabel Rata-rata Tingkat Aktivitas Harian (di luar latihan)

Tingkat Aktivitas	Jenis Kelamin	
	Laki-laki	Perempuan
Istirahat di tempat tidur	1,2	1,2
Kerja sangat ringan	1,4	1,4
Kerja ringan	1,5	1,5
Kerja ringan-sedang	1,7	1,6
Kerja sedang	1,8	1,7
Kerja berat	2,1	1,8
Kerja berat sekali	2,3	2,0

Tabel 2.4 : Tabel Kebutuhan Energi Aktivitas Olahraga Berdasarkan Berat Badan (Kal/menit)

Aktivitas	Berat Badan (Kg)				
	50	60	70	80	90
(1)	(2)	(3)	(4)	(5)	(6)
Sepakbola					
kipper	7	8	9	10	12
defender	8	9	11	12	13
striker	9	10	11	13	14
Lari :					
- 5,5 menit/Km	10	12	14	15	17

- 5 menit/Km	10	12	15	17	19
- 4,5 menit/Km	11	13	15	18	20
- 4 menit/Km	13	15	18	21	23
Jalan Kaki :					
- 10 menit/Km	5	6	7	8	9
- 8 menit/Km	6	7	8	10	11
- 5 menit/Km	10	12	15	17	19

d. Pertumbuhan

Anak dan remaja mengalami pertumbuhan sehingga memerlukan penambahan energi. Energi tambahan diperlukan untuk pertumbuhan tulang baru dan jaringan tubuh.

Tabel 2.5 : Tabel Kebutuhan Energi Untuk Pertumbuhan (Kal/hari)

Umur (tahun)	Tambahan Energi
10 – 14	2 Kal/Kg berat badan
15	1 Kal/Kg berat badan
16 – 18	0,5 Kal/Kg berat badan

2.1.2 Perhitungan Kebutuhan Gizi Remaja

a. Kalori (Energi)

BMR + SDA + Aktivitas Fisik

Dijabarkan dalam 6 langkah dalam menghitung kebutuhan energi yaitu :

1. Menghitung indeks massa tubuh (IMT)

$$\text{IMT} = A / B^2$$

Keterangan :

IMT = indeks massa tubuh

A = berat badan (kg)

B = tinggi badan (m)

Tabel 2.6 Tabel Ukuran status gizi menurut IMT

IMT	KATEGORI
< 17,00	Kurus sekali
17,0 – 18,4	Kurus
18,5 – 25,0	Normal
25,1 – 27,0	Gemuk
> 27,0	Gemuk sekali

Bila berat badan dinilai kurang dari berat badan ideal, maka kebutuhan energinya ditambah sebanyak 500 kkalori, sedangkan bila lebih, dikurangi sebanyak 500 kkalori dalam sehari.

2. Tentukan BMR yang sesuai dengan jenis kelamin, umur dan berat badan seperti pada table 1 dan 2. Tambahkan BMR dengan SDA yang besarnya 10% BMR.

$$\text{BMR} + \text{SDA (10 \% BMR)}$$

3. Tentukan faktor tingkat aktivitas fisik setiap hari (tanpa olahraga) sesuai table 3.

4. Kalikan hasil dari langkah 2 dengan faktor tingkat aktivitas fisik yang tertera pada table 3.
5. Tentukan penggunaan energi sesuai dengan latihan atau pertandingan sepakbola dengan menggunakan tabel 4. Kalikan jumlah jam yang digunakan untuk latihan per minggu dengan besar energi yang dikeluarkan untuk setiap latihan olahraga. Total perhitungan energi yang didapat dari perhitungan dalam seminggu, kemudian dibagi 7 untuk mendapatkan penggunaan energi yang dikeluarkan per hari. Tambahkan besarnya penggunaan energi ini dengan besarnya energi yang didapatkan pada perhitungan langkah 4.
6. Apabila atlet tersebut dalam usia pertumbuhan, maka tambahkan kebutuhan energi sesuai tabel 5.

b. Protein

Rumus yang digunakan untuk menghitung kebutuhan protein adalah :

$$\frac{\text{Kebutuhan_kalori}}{10} \times 0,25$$

c. Lemak

Rumus yang digunakan untuk menghitung kebutuhan lemak adalah

$$25\% \times \text{kebutuhan kalori}$$

d. Karbohidrat

Rumus yang digunakan untuk menghitung kebutuhan karbohidrat adalah : (Direktorat Gizi masyarakat. 2002: 3-8)

$$70\% \times \text{kebutuhan kalori}$$

2.2 Bahan pangan

Penggolongan makanan menurut DKBM daftar komposisi bahan Makanan. Sumber LIPI 1988. (Kusumadewi Sri,2007:234)

Tabel 2.7 Tabel golongan bahan pangan

Golongan	Bahan Pangan
1	Padi-padian
2	Umbi-umbian
3	Pangan hewani
4	Kacang-kacangan
5	Buah-biji berminyak
6	Minyak dan lemak
7	Gula
8	Sayur dan buah
9	Lain-lain

2.3 Optimasi

Dalam kehidupan sehari-hari, baik disadari maupun tidak, orang selalu melakukan optimasi untuk memenuhi kebutuhannya. Optimasi merupakan aktivitas untuk mendapatkan hasil yang terbaik dari pilihan yang tersedia. Teknik optimasi merupakan salah satu bagian dari program matematika yang sudah banyak diaplikasikan. Tujuan dari setiap keputusan adalah untuk meminimumkan usaha yang dilakukan atau memaksimalkan keuntungan yang diperoleh. Usaha atau keuntungan tersebut secara praktek dinyatakan sebagai fungsi dengan variabel keputusan yang akan dicari nilai optimumnya.

Optimasi yang dilakukan oleh masyarakat awam lebih banyak dilandasi oleh intuisi daripada teori optimasi. Dalam bidang rekayasa optimasi sangat dibutuhkan, sering dihadapkan pada persoalan mencari penyelesaian termurah dengan memenuhi segala kendala yang ada.

Untuk memiliki teknologi optimasi, seorang perencana perlu mendalami teknik-teknik optimasi baik yang sederhana untuk mendapatkan pengertian mendasar maupun yang canggih untuk menyelesaikan permasalahan nyata di lapangan. Topik mengenai optimasi di negara-negara berkembang merupakan bidang keahlian tersendiri yang membutuhkan waktu yang tidak sedikit untuk mendalaminya. Riset-riset yang terkait dengan masalah optimasi masih terus berlanjut sampai sekarang sehingga banyak temuan teknik baru yang lebih canggih dan efisien. (Kusumadewi,Sri. 2005: 34)

2.4 Algoritma Genetika

2.4.1 Pengertian

Algoritma adalah inti dari informatika. Dan algoritma yang baik akan menjamin kualitas dari informasi yang dihasilkan. Algoritma sudah banyak diterapkan sebagai penyelesaian dalam berbagai permasalahan kehidupan termasuk penyelesaian dalam persoalan agama juga persoalan ilmu pengetahuan lainnya seperti proses biologi, fisika dan kimia. Sedangkan dalam bidang informatika telah berkembang istilah kecerdasan buatan dimana terdapat metode-metode komputasi yang mempunyai toleransi terhadap ketidakpastian. Metode ini banyak meniru sifat-sifat alam seperti cara kerja

manusia, binatang atau proses fisika di alam semesta. Salah satu metode tersebut adalah algoritma genetika yang oleh penulis digunakan untuk membantu proses optimasi agar didapatkan hasil yang optimal.

Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional (Anita dan Muhammad, 2006:187)

Algoritma genetika adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Pada definisi tersebut, terdapat dua istilah yang sudah tidak asing lagi bagi pembaca yang pernah mempelajari teori evolusi. Istilah seleksi alamiah secara sederhana dilustrasikan dalam kasus populasi jerapah. Sedangkan genetika alamiah adalah mekanisme yang sangat rumit, sampai saat ini ilmu pengetahuan hanya dapat menjelaskan hal tersebut secara sederhana sebagai berikut. Dalam ilmu biologi, sekumpulan individu yang sama (yang disebut *species*) hidup, bereproduksi, dan mati dalam satu area yang disebut populasi. Jika anggota-anggota populasi (individu) terpisah, misalkan karena bencana alam, maka individu-individu tersebut membentuk beberapa populasi yang terpisah. Dalam waktu yang cukup lama, mungkin saja akan terjadi proses pembentukan *species* baru atau dikenal dengan istilah *speciation* (Suyanto, 2007: 205-206).

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap

hidup. Pada dasarnya ada empat kondisi yang sangat mempengaruhi proses evaluasi, yaitu:

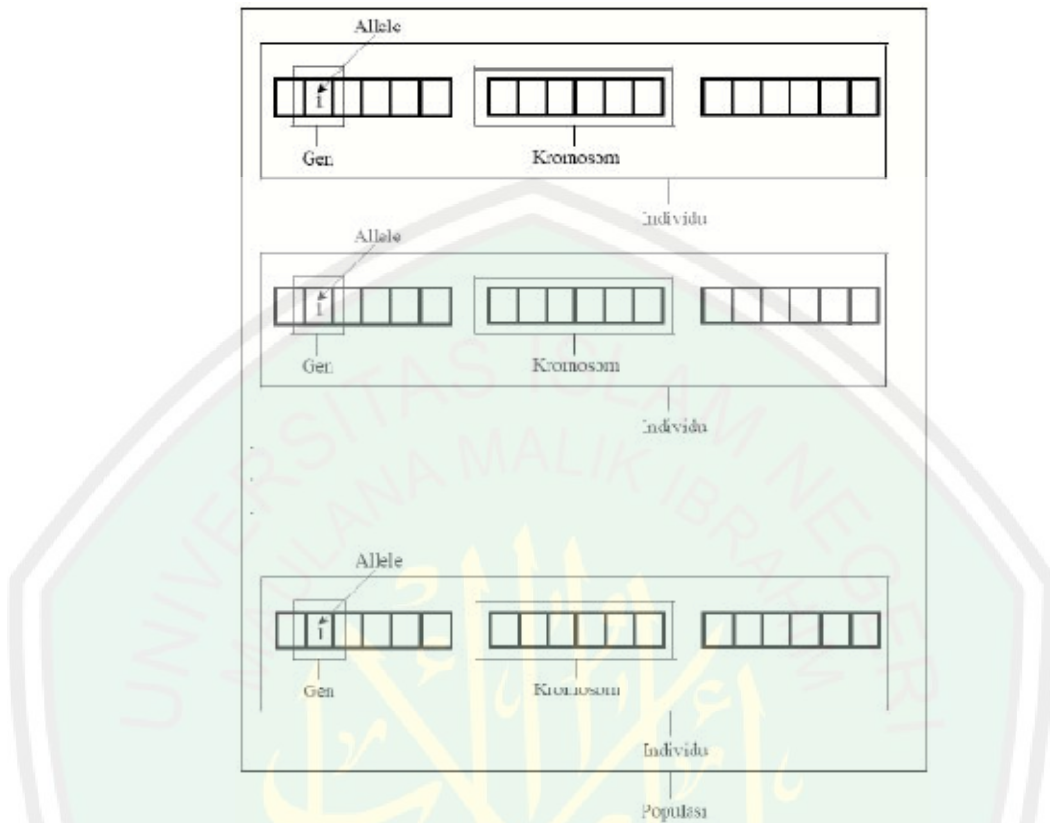
- a. Kemampuan organisme untuk melakukan reproduksi.
- b. Keberadaan populasi organisme yang bisa melakukan reproduksi.
- c. Keberagaman organisme dalam suatu populasi.
- d. Perbedaan kemampuan untuk *survive*.

Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (Kusumadewi, Sri. 2003: 279).

Ada beberapa istilah yang perlu dimengerti dalam algoritma genetika, di antaranya :

- a. Kromosom: individu yang terdapat dalam satu populasi. Kromosom ini merupakan solusi yang masih berbentuk simbol.
- b. Individu: menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- c. *Genotype* (Gen): sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, *float*, *integer* maupun karakter.
- d. *Allele*: nilai yang berada dalam gen.
- e. *Locus*: letak suatu gen berada dalam suatu kromosom.

- f. Fungsi Fitness: alat ukur dalam dalam proses evaluasi yang dilalui kromosom. Nilai fitness akan menunjukkan kualitas dari suatu kromosom dalam populasi tersebut.
- g. *Offspring*: anak (generasi berikutnya) yang terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*) maupun operator mutasi.
- h. Populasi: sejumlah solusi yang mungkin. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan generasi.
- i. Generasi: iterasi yang dilakukan untuk menentukan populasi berikutnya.

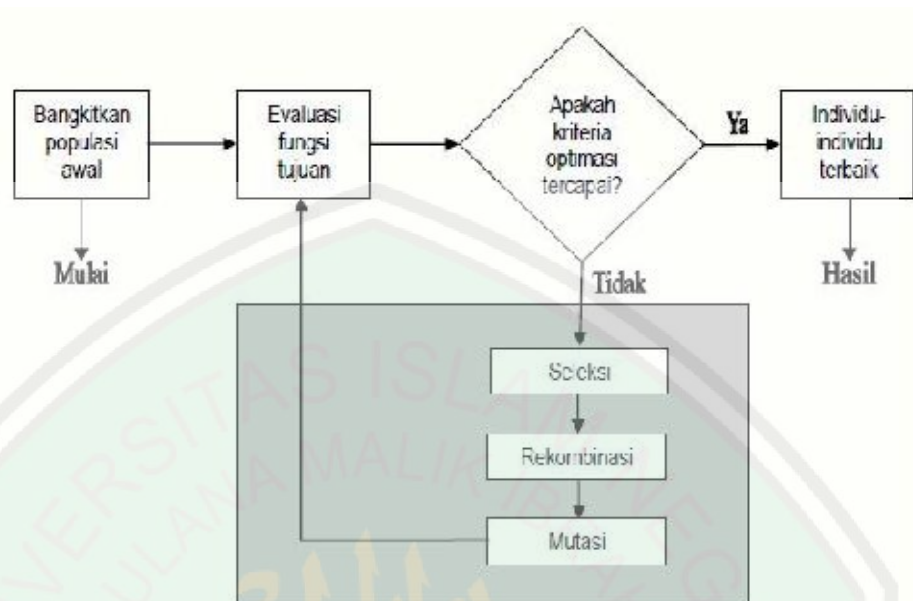


Gambar 2.1 Ilustrasi istilah pada Algoritma Genetika

Secara umum sebuah penerapan algoritma genetika akan melalui siklus sederhana yang terdiri dari 4 langkah, yaitu:

- Membangun sebuah “populasi” yang terdiri dari beberapa string.
- Evaluasi masing-masing *string* (*fitness value*).
- Proses seleksi agar didapat *string* yang terbaik.
- Manipulasi genetika untuk menciptakan populasi baru dari *string*.

Secara sederhana, siklus algoritma genetika dapat terlihat seperti gambar berikut:



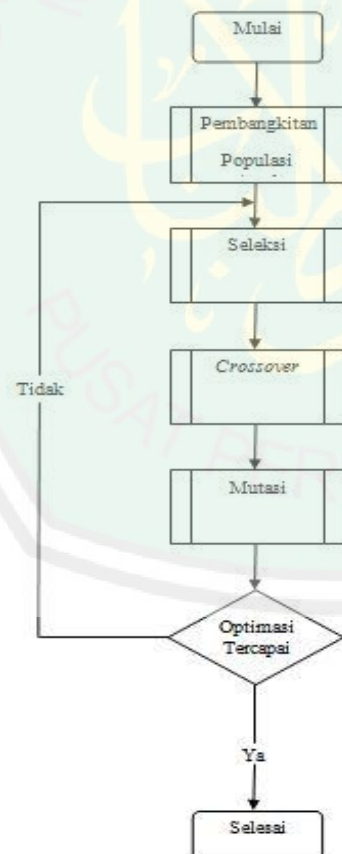
Gambar 2.2 Siklus Sederhana Algoritma Genetika

2.4.2 Struktur Algoritma Genetika

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator

penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk (*parent*) dan nilai fitness dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik (Sri, 2003: 280).

Secara umum, algoritma genetika dapat diilustrasikan dalam diagram alir sebagai berikut:



Gambar 2.3 *flowchart* algoritma genetika

2.4.3 Komponen Utama Algoritma Genetika

Ada enam komponen utama dalam algoritma genetika, yaitu:

a. Teknik Pengkodean

Langkah pertama pada algoritma genetika adalah menerjemahkan atau merepresentasikan masalah riil menjadi terminologi biologi. Cara untuk merepresentasikan masalah ke dalam bentuk kromosom disebut pengkodean. Terdapat beberapa cara pengkodean, dan pemilihannya berdasarkan masalah yang dihadapi. Teknik pengkodean disini meliputi pengkodean gen dari kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk: *string* bit, pohon, *array* bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator algoritma genetika. Demikian juga kromosom dapat direpresentasikan dengan menggunakan:

- *String* bit : 10011, 01101, 11101, dst.
- Bilangan real : 65.65, -67.98, 562.88, dst.
- Elemen permutasi : E2, E10, E5, dst.
- Daftar aturan : R1, R2, R3, dst.
- Elemen program : pemrograman genetika
- Struktur lainnya.

Berikut adalah beberapa jenis pengkodean yang umum digunakan.

1) Pengkodean Biner

Pengkodean biner adalah pengkodean yang paling umum dan paling sederhana dalam merepresentasikan masalah pada algoritma genetika.

Pada pengkodean biner setiap kromosom terdiri atas barisan string bit 0 atau 1. Contoh masalah yang sesuai menggunakan pengkodean biner adalah masalah nilai maximize pada sebuah fungsi matematika.

2) Pengkodean Permutasi

Pengkodean permutasi dapat digunakan pada masalah pengurutan data (*ordering problems*), seperti wiraniaga (*Travelling Salesman Problem*) atau masalah pengurutan tugas (*Task Ordering Problem*). Pada pengkodean permutasi, setiap kromosom terdiri dari barisan angka, yang merepresentasikan angka pada urutan.

3) Pengkodean Nilai

Pengkodean nilai dapat digunakan pada masalah yang sangat kompleks, dimana nilai yang dikodekan langsung merupakan representasi dari masalah. Penggunaan pengkodean biner pada tipe masalah yang kompleks akan menjadi lebih susah. Pada pengkodean nilai, setiap kromosom adalah barisan dari beberapa nilai. Nilai dapat berupa apa saja, seperti bilangan biasa, bilangan riil, karakter sampai dengan objek-objek yang rumit.

4) Pengkodean Pohon

Pengkodean pohon lebih banyak digunakan untuk menyusun program atau ekspresi, bagi pemrograman genetika (*genetic programming*). Pada pengkodean pohon, setiap kromosom merupakan pohon dari sejumlah obyek, seperti fungsi atau perintah pada bahasa pemrograman.

b. Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

c. Fungsi Evaluasi

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi obyektif (fungsi tujuan) dan konversi fungsi obyektif dengan nilai yang tidak negatif. Apabila ternyata fungsi obyektif memiliki nilai negatif, maka perlu ditambahkan suatu konstanta C agar nilai fitness yang terbentuk menjadi tidak negatif.

d. Seleksi

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Ada beberapa metode seleksi dari induk, antara lain:

- *Rank-based fitness assignment*

- *Roulette wheel selection*
- *Stochastic universal sampling*
- *Local selection*
- *Truncation selection*
- *Tournament selection*

e. Operator Genetika

Ada dua operator genetika, yaitu:

1) Operator untuk melakukan rekombinasi, yang terdiri dari:

a. Rekombinasi bernilai real

- i. Rekombinasi *diskret*
- ii. Rekombinasi *intermediate* (menengah)
- iii. Rekombinasi garis
- iv. Rekombinasi garis yang diperluas

b. Rekombinasi bernilai biner (*crossover*)

- i. *Crossover* satu titik
- ii. *Crossover* banyak titik
- iii. *Crossover* seragam
- iv. *Crossover* dengan permutasi

c. Mutasi

- i. Mutasi bernilai real
- ii. Mutasi bernilai biner

f. Penentuan Parameter

Yang disebut dengan parameter di sini adalah parameter kontrol Algoritma genetika, yaitu: ukuran populasi (*popsize*), peluang crossover (*Pc*), dan peluang mutasi (*Pm*). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. Ada beberapa rekomendasi yang bisa digunakan, antara lain (Sri, 2003: 283):

- A. Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan untuk nilai parameter kontrol: (*Popsize*; *Pc*; *Pm*) = (50; 0,6; 0,001)
- B. Bila rata-rata fitness setiap generasi digunakan sebagai indikator, maka Grefensette merekomendasikan: (*Popsize*; *Pc*; *Pm*) = (30; 0,95; 0,01)
- C. Bila fitness dari individu terbaik dipantau pada setiap generasi maka usulannya adalah: (*Popsize*; *Pc*; *Pm*) = (80; 0,45; 0,01)
- D. Ukuran populasi sebaiknya tidak lebih kecil dari 30, untuk sembarang jenis permasalahan.

2.4.4 Nilai Fitness

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam *evolusi* alam, individu yang bernilai fitness tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai fitness rendah akan mati.

Pada masalah optimasi, jika solusi yang dicari adalah memaksimalkan sebuah fungsi f (dikenal sebagai masalah maksimasi),

maka nilai *fitness* yang digunakan adalah nilai dari fungsi h tersebut, yakni $f = h$ (di mana f adalah nilai *fitness*). Tetapi jika masalahnya adalah meminimalkan fungsi h (masalah minimasi), maka fungsi h tidak bisa digunakan secara langsung. Hal ini disebabkan adanya aturan bahwa individu yang memiliki nilai *fitness* tinggi lebih mampu bertahan hidup pada generasi berikutnya. Oleh karena nilai *fitness* yang bisa digunakan adalah $f = 1 / h$, yang artinya semakin kecil nilai h , semakin besar nilai f (Suyanto, 2006: 10-11). Tetapi hal ini akan menjadi masalah jika h bisa bernilai 0, yang mengakibatkan f bisa bernilai tak hingga. Untuk mengatasinya, h perlu ditambah sebuah bilangan yang dianggap sangat kecil sehingga nilai *fitness*nya menjadi: $f = 1 / (h + a)$, di mana a adalah bilangan yang dianggap sangat kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan.

2.4.5 Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit (Anita dan Muhammad, 2006: 193). Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang

nantinya akan digunakan pada tahap-tahap seleksi berikutnya (Kusumadewi,Sri dan Hari, 2005: 235).

Kemampuan algoritma genetika untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikutnya. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik berhubungan dengan penekanan selektif yang diterapkan pada individu anak tersebut.

Cara lain yaitu menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sebanyak yang dipelihara dalam populasi yang perlu dibuat bagi generasi berikutnya. Walaupun penekanan selektif tidak diterapkan ke level keturunan, metode ini akan terus menghasilkan kromosom yang lebih baik, karena adanya penekanan yang diterapkan ke orang tua (Anita dan Muhammad, 2006:193-194).

a. *Rank-based Fitness*

Pada rank-based fitness, populasi diurutkan menurut nilai objektifnya. Nilai fitness dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya.

Proses dimulai dengan meranking atau mengurutkan kromosom di dalam populasi berdasarkan fitness-nya kemudian memberi nilai fitness baru berdasarkan urutannya. Kromosom dengan nilai terburuk akan memiliki fitness baru nilai 1, terburuk kedua bernilai 2 dan begitu seterusnya, sehingga kromosom yang memiliki fitness terbaik akan memiliki nilai fitness N , dimana N adalah jumlah kromosom di dalam populasi. Setelah adanya proses seleksi tersebut, maka saat ini seluruh kromosom mempunyai kesempatan untuk dipilih. Akan tetapi, metode ini dapat menyebabkan konvergensi menjadi lambat, karena kromosom terbaik tidak terlalu berbeda dengan yang lainnya.

b. *Seleksi Roda Roulette*

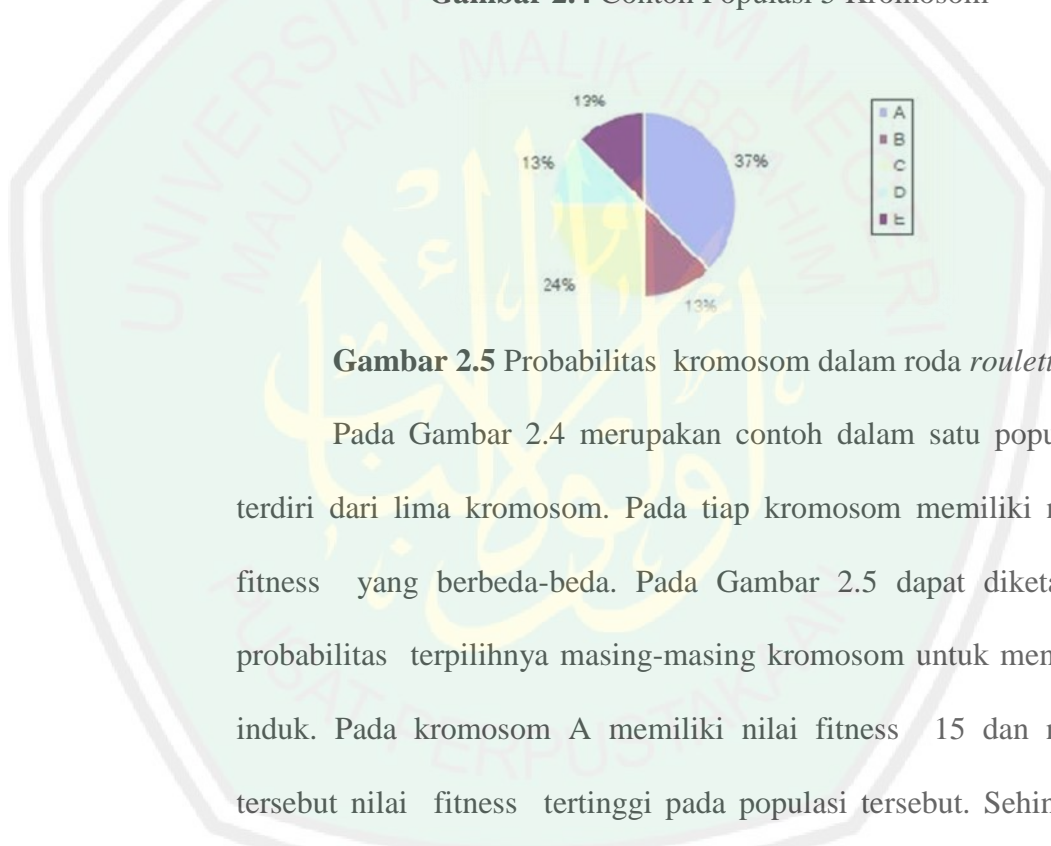
Metode seleksi roda *roulette* merupakan metode yang paling sederhana, dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran fitness-nya.

Pada metode ini induk dipilih berdasarkan nilai fitnessnya, semakin besar nilai fitness maka akan semakin besar kemungkinannya untuk terpilih menjadi induk. Diandaikan semua kromosom diletakkan pada sebuah roda roulette, besarnya

kemungkinan bagi setiap kromosom adalah tergantung dari nilai fitness-nya seperti pada contoh berikut:

Kromosom	<i>Fitness</i>
A	15
B	5
C	10
D	5
E	5

Gambar 2.4 Contoh Populasi 5 Kromosom



Gambar 2.5 Probabilitas kromosom dalam roda *roulette*

Pada Gambar 2.4 merupakan contoh dalam satu populasi terdiri dari lima kromosom. Pada tiap kromosom memiliki nilai fitness yang berbeda-beda. Pada Gambar 2.5 dapat diketahui probabilitas terpilihnya masing-masing kromosom untuk menjadi induk. Pada kromosom A memiliki nilai fitness 15 dan nilai tersebut nilai fitness tertinggi pada populasi tersebut. Sehingga kromosom A memiliki probabilitas terbesar untuk terpilih menjadi induk.

c. *Stochastic Universal Sampling*

Stochastic universal sampling memiliki nilai bias nol dan penyebaran yang minimum. Metode ini seperti halnya pada metode seleksi roda roulette, hanya saja diberikan sejumlah pointer sebanyak individu yang ingin diseleksi di pada garis tersebut. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak antar pointer adalah $1/N$, dan posisi pointer pertama diberikan secara acak pada range $[1,1/N]$.

d. Seleksi Lokal

Pada seleksi lokal, setiap individu yang berada di dalam konstrain tertentu disebut dengan nama lingkungan lokal. Interaksi antar individu hanya dilakukan di dalam wilayah tersebut. Lingkungan tersebut ditetapkan sebagai struktur di mana populasi tersebut terdistribusi. Lingkungan tersebut juga dapat dipandang sebagai kelompok pasanganpasngan yang potensial.

e. Seleksi dengan Pematangan

Pada metode-metode yang telah dijelaskan sebelumnya, seleksi dilakukan secara alami. Pada seleksi dengan pematangan ini, lebih berkesan sebagai seleksi buatan. Seleksi ini biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu-individu diurutkan berdasarkan berdasarkan nilai fitnessnya. Hanya individu-individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam

metode ini adalah suatu nilai ambang trunk yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50%-100%. Individu-individu yang ada di bawah nilai ambang ini tidak akan menghasilkan keturunan.

f. Seleksi dengan Turnamen

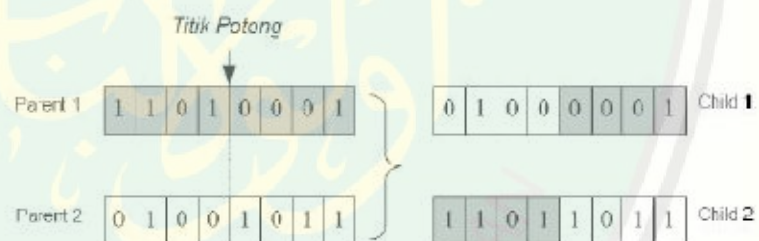
Pada metode seleksi dengan turnamen ini, akan ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara random dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

Seleksi turnamen merupakan variasi antara seleksi roda roulette dan seleksi ranking. Sejumlah k kromosom tertentu dari populasi beranggota n kromosom ($k \leq n$) dipilih secara acak dengan probabilitas yang sama. Dari k kromosom yang terpilih kemudian akan dipilih satu kromosom dengan fitness terbaik, yang diperoleh dari hasil pengurutan ranking fitness semua kromosom terpilih. Perbedaan dengan seleksi roda roulette adalah pemilihan kromosom yang akan digunakan untuk berkembangbiak tidak berdasarkan skala fitness dari populasi.

2.4.6 Crossover

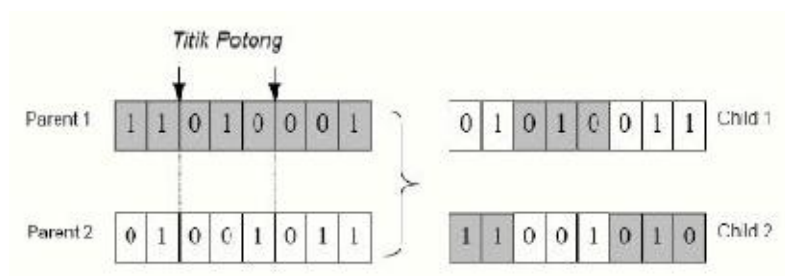
Crossover adalah suatu proses untuk menyilangkan dua buah kromosom sehingga didapat kromosom yang mempunyai karakteristik yang mirip dengan induknya. Proses *crossover* dilakukan agar kromosom yang dianggap tidak pantas untuk dipertahankan pada generasi selanjutnya bisa digantikan dengan kromosom yang lebih baik dari induknya. Banyaknya kromosom yang akan mengalami proses *crossover* sesuai dengan probabilitas yang telah ditetapkan terlebih dahulu. Jumlah kromosom yang akan di-*crossover* harus berjumlah genap karena proses *crossover* memerlukan sepasang kromosom untuk digabungkan sifatnya.

a. *Crossover* satu titik



Gambar 2.6 *crossover* 1 titik

b. *Crossover* dua titik



Gambar 2.7 *crossover* 2 titik

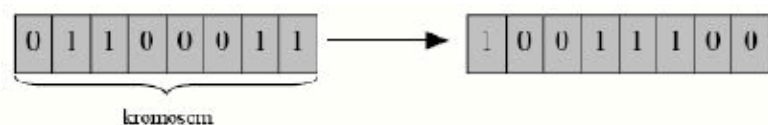
2.4.7 Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inialisasi populasi. Mutasi diterapkan dengan probabilitas sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi bit pada kromosom yang unggul akan dirusak. Mutasi ini bukanlah operator yang utama, yang dilakukan secara acak pada gen dengan kemungkinan yang kecil. (P_m sekitar 0,001) (Son,2007: 185).

Peluang mutasi (P_m) mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi terlalubesar, maka akan terlalu banyak gangguan acak, sehingga akan kehilangan kemiripan dari induknya.

Berdasarkan bagian yang termutasi, proses mutasi dapat dibedakan atas tiga bagian:

- a. Tingkat kromosom : semua gen dalam kromosom berubah.



Gambar 2.8 mutasi tingkat kromosom

- b. Tingkat gen : Misal gen 2 yang mengalami mutasi.



Gambar 2.9 mutasi tingkat gen

- c. Tingkat bit : hanya satu bit yang berubah



Gambar 2.10 mutasi tingkat bit

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Algoritma genetika yang memiliki kehandalan dalam menghasilkan output yang optimal dapat dimanfaatkan untuk memecahkan masalah tersebut dengan bantuan piranti lunak. Pada piranti lunak yang dibuat, terdapat beberapa input yang dibutuhkan, yaitu jumlah kromosom awal, jumlah generasi, probabilitas *crossover* dan probabilitas mutasi. Hasil pemrosesan merupakan kombinasi makanan yang merepresentasikan penyelesaian masalah ini.

Namun sebelum melakukan proses algoritma genetika perlu diketahui status gizi dari atlet sepakbola tersebut. Karena proses optimasi yang menggunakan algoritma genetika ini, bertujuan mendapatkan komposisi bahan pangan harian atlet sepakbola yang memenuhi kecukupan gizi, jadi selain juga diperlukan data bahan pangan beserta kandungannya.

3.1.1 Pembangkitan Populasi Awal

Populasi disini sebagai kumpulan solusi yaitu tentunya komposisi bahan pangan yang diharapkan. Populasi awal terdiri dari kromosom-kromosom. Kromosom tersebut terdiri dari gen-gen dimana gen tersebut merupakan *encoding* yang digunakan dalam merepresentasikan masalah tersebut. Teknik *encoding* yang digunakan pada persoalan ini adalah *value encoding*. Pada

value encoding, setiap bahan pangan diwakili oleh sebuah angka, dan angka tersebut merupakan sebuah gen yang ada pada sebuah kromosom.

Setiap kromosom yang ada pada satu generasi merepresentasikan komposisi bahan pangan yang digunakan untuk mendapatkan hasil yang optimal, dan kromosom yang mewakili solusi tersebut boleh mempunyai lebih dari satu jenis makanan yang sama.

Seperti dijelaskan sebelumnya di dalam kromosom terdapat gen. Gen disini adalah kode kode dari bahan pangan. Pengelompokan bahan pangan menurut daftar komposisi bahan makanan atau disebut DKBM. pada bab sebelumnya sudah sedikit dijelaskan, dan berikut perinciannya

Tabel 3.1 Kelompok 1 Padi-padian dan hasil olahannya

Bahan Pangan	Energi (kal)	Protein (gram)
a. Beras giling	360	6,8
b. Beras merah	359	7,5
c. Terigu	365	0,9
d. Roti putih	248	8,0
e. Biscuit	458	6,9
f. Bihun	360	4,7

Tabel 3.2 Kelompok 2 Umbi-umbian dan hasil olahannya

Bahan Pangan	Energi (kal)	Protein (gram)
a. Kentang	83	2,0
b. Ubi jalar	123	1,8
c. Sagu	353	0,7

Tabel 3.3 Kelompok 3 pangan hewani dan hasil olahannya

Bahan Pangan	Energi (kal)	Protein (gram)
a. Daging ayam	302	18,2
b. Daging bebek	326	16,0
c. Daging sapi	207	18,8
d. Daging kambing	154	16,6
e. Telur ayam	162	12,8
f. Telur bebek	189	13,1
g. Ikan bandeng	129	20,0
h. Ikan mas	86	16,0
i. Ikan kakap	92	20,0
j. Udang	91	21,0

Tabel 3.4 Kelompok 4 Kacang-kacangan dan hasil olahannya

Bahan pangan	Energi (kal)	Protein (gram)
a. Kacang bogor	370	16,0
b. Kacang hijau	345	22,2
c. Kacang kedelai	286	30,2
d. Kacang merah	336	23,1
e. Oncom	187	13,0
f. Tahu	68	7,8
g. Tempe	149	18,3

Tabel 3.5 Kelompok 5 buah-biji berminyak dan hasil olahannya

Bahan pangan	Energi (kal)	Protein (gram)
a. Kelapa	720	0,6
b. Alpukat	902	0,0
c. Minyak kelapa	870	1,0

Tabel 3.6 Kelompok 6 minyak, lemak dan hasil olahannya

Bahan pangan	Energi (kal)	Protein (gram)
a Margarin	720	0,6
b Minyak ikan	902	0,0
c Minyak kelapa	870	1,0

Tabel 3.7 Kelompok 7 gula dan hasil olahannya

Bahan pangan	Energi (kal)	Protein (gram)
a. Gula pasir	364	0
b. Madu	294	0,3
c. Dodol	395	3,0
d. Teh	132	19,5
e. Coklat susu	381	9,0

Tabel 3.8 Kelompok 8 sayur dan buah

Bahan pangan	Energi (kal)	Protein (gram)
a. Bayam	36	3,5
b. Kangkung	29	3,0
c. Sawi	22	2,3
d. Wortel	42	0,0
e. Selada	15	1,2
f. Kacang panjang	44	2,7
g. Daun melinjo	99	5,0
h. Alpokat	85	0,9
i. Apel	58	0,3
j. Belimbing	36	0,4
k. Papaya	99	1,2
l. Jeruk manis	45	0,9

Tabel 3.9 Kelompok 9 lain-lain

Bahan pangan	Energi (kal)	Protein (gram)
a. Es krim	201	4,0
b. Susu sapi	61	3,2
c. Yoghurt	52	3,3
d. Susu krim	36	3,5

Inisialisasi kromosom

Kromosom 1

1a	2c	3d	4b	5a	6d	7b	8c	9a	fitnes
----	----	----	----	----	----	----	----	----	--------

Kromosom 2

1c	2a	3b	4d	5d	6a	7c	8a	9b	fitnes
----	----	----	----	----	----	----	----	----	--------

Kromosom 20

1g	2d	3e	4a	5c	6c	7b	8e	9c	fitnes
----	----	----	----	----	----	----	----	----	--------

Setiap kromosom terdapat 9 gen yang dipilih secara random mewakili kelompok bahan pangan. Dan terdapat 20 kromosom mengikuti penelitian sebelumnya, namun jumlah kromosom bisa dirubah seperti maksimum generasi. Juga terdapat fitness yang mewakili setiap kromosom, dimana nilai fitness suatu kromosom akan menunjukkan kualitas kromosom dalam populasi.

3.1.2 Fungsi fitness

Individu-individu dalam populasi telah terbentuk, maka langkah selanjutnya adalah proses evaluasi. Proses evaluasi disini dilakukan dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari

suatu kromosom akan menunjukkan kualitas kromosom dalam populasi.

Fungsi fitness yang digunakan adalah:

$$f = \frac{1}{((ab(m - \sum a) + ab(n - \sum b)) + bil\ kecil)}$$

Keterangan:

m = kebutuhan kalori keseluruhan yang dihitung secara manual

n = kebutuhan protein selama 1 hari yang dihitung secara manual

a = jumlah kandungan kalori dari 1 bahan makanan

b = jumlah kandungan protein dari 1 bahan makanan

bilKecil = bilangan untuk menghindari pembagian dengan nol

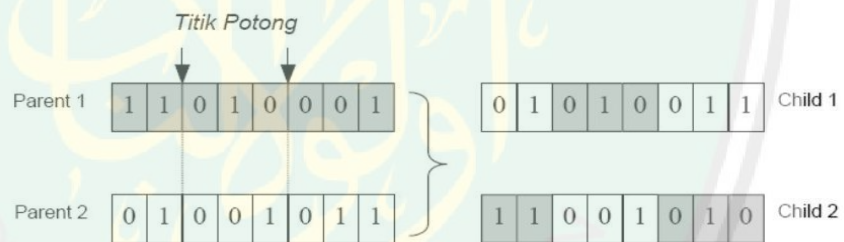
3.1.3 Proses Seleksi

Setelah melalui proses evaluasi, tahap selanjutnya adalah melakukan seleksi. Metode seleksi yang digunakan disini adalah metode seleksi roda roulette (*roulette wheel*) yaitu Calon induk yang akan dipilih berdasarkan nilai *fitness* yang dimilikinya, semakin baik individu tersebut yang ditunjukkan dengan semakin besar nilai fitnessnya akan mendapatkan kemungkinan yang lebih besar untuk terpilih sebagai induk.

Misalkan saja *roulette wheel* merupakan tempat untuk menampung seluruh kromosom dari tiap populasi, maka besarnya tempat dari *roulette wheel* tersebut menunjukkan seberapa besar nilai *fitness* yang dimiliki oleh suatu kromosom, semakin besar nilai *fitness* tersebut, maka semakin besar pula tempat yang tersedia.

3.1.4 Proses Kawin Silang (*Crossover*)

Crossover adalah suatu proses untuk menyilangkan dua buah kromosom sehingga didapat kromosom yang mempunyai karakteristik yang mirip dengan induknya. Proses *crossover* dilakukan agar kromosom yang dianggap tidak pantas untuk dipertahankan pada generasi selanjutnya bisa digantikan dengan kromosom yang lebih baik dari induknya. Banyaknya kromosom yang akan mengalami proses *crossover* sesuai dengan probabilitas yang telah ditetapkan terlebih dahulu. Jumlah kromosom yang akan di-*crossover* harus berjumlah genap karena proses *crossover* memerlukan sepasang kromosom untuk digabungkan sifatnya.



Gambar 3.1 *Crossover* dua titik

Pemilihan gen yang akan di-*crossover* dilakukan berdasarkan probabilitas *crossover*. Nilai probabilitas *crossover* tersebut dikalikan dengan jumlah kromosom yang ada. Hasil kalinya adalah jumlah kromosom yang akan di-*crossover*. Langkah selanjutnya adalah membangkitkan bilangan acak sebanyak jumlah kromosom. Apabila bilangan acak kurang dari probabilitas *crossover*, maka pada kromosom yang bersesuaian akan dilakukan proses *crossover*. Pada aplikasi ini, teknik *crossover* yang digunakan adalah *crossover* dua titik. Operator pindah silang dapat dilakukan dengan lebih

dari dua titik. Tetapi jumlah titik potong yang semakin banyak akan memperendah kualitas solusi yang didapatkan. Hal ini disebabkan operasi pindah silang terlalu sering merusak kromosom yang baik.

3.1.5 Proses Mutasi

Proses mutasi merupakan proses dimana suatu gen akan mengalami penyimpangan dari kromosom induknya sehingga sifat kromosom anak tersebut akan mengalami perbedaan dari kromosom induknya. Banyaknya jumlah kromosom yang akan dimutasi tergantung dari probabilitas yang telah ditentukan nilainya.

Proses mutasi dilakukan dengan cara mengkalikan probabilitas mutasi dengan jumlah gen dan kromosom dalam suatu populasi. Langkah selanjutnya adalah membangkitkan bilangan acak sebanyak perkalian jumlah kromosom dengan jumlah gen. Proses selanjutnya adalah membandingkan bilangan acak dengan nilai probabilitas mutasi. Apabila bilangan acak tersebut lebih kecil dari nilai probabilitas mutasi, maka gen tersebut akan mengalami mutasi.

Pada kasus ini skema mutasi yang digunakan adalah *swapping mutation/mutation exchange*. Gen yang dimutasi nilainya ditukar dengan salah satu nilai yang ada di tabel makanan dan belum masuk pada daftar nilai yang ada pada kromosom tersebut. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation rate* (P_m) atau peluang mutasi. Peluang mutasi mengendalikan banyaknya gen baru

yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma akan kehilangan kemampuan untuk belajar dari histori pencarian.

Mutasi dilakukan untuk mencegah terjadinya konvergensi prematur. Pada mutasi cara *swap* (Penukaran) adalah mutasi yang dilakukan dengan menukar langsung nilai dari gen. Pemilihan cara mutasi dilakukan secara *random*.

3.2 Perancangan Sistem

Setelah melakukan analisis terhadap data yang terkumpul selanjutnya dapat dilakukan perancangan sistem dari sistem manual kedalam metode Algoritma Genetika (*Genetic Algorithm*).



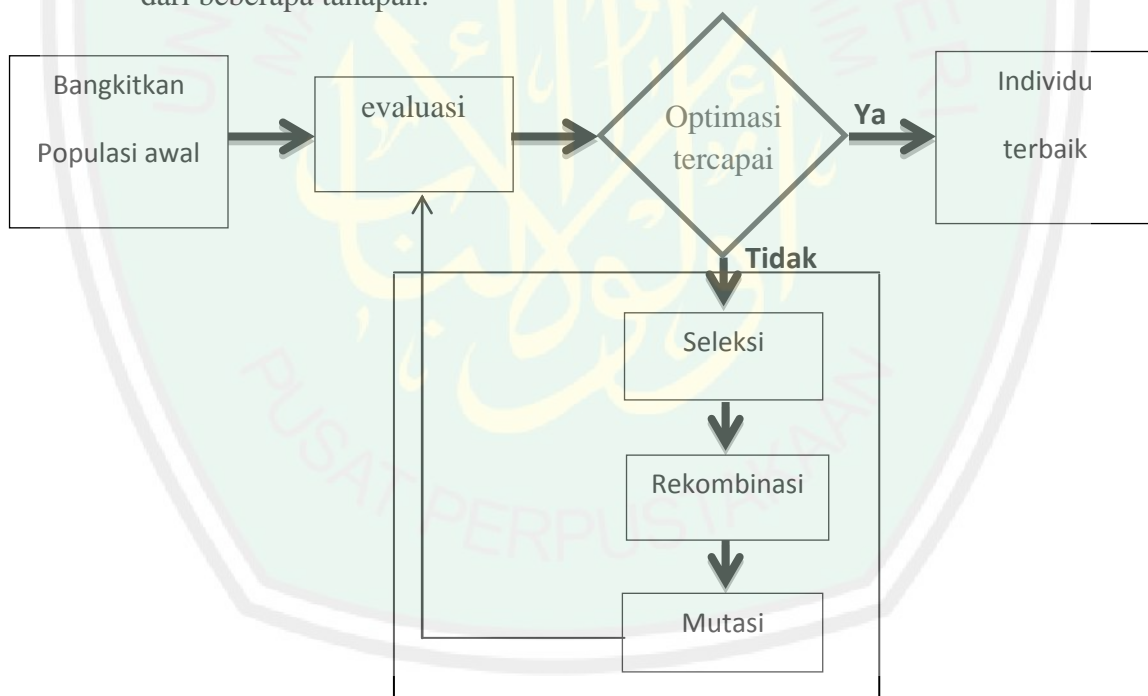
Gambar 3.2 input-proses-output

Dari gambar di atas dapat dilihat bahwa proses optimasi Penentuan bahan pangan harian atlet sepakbola dengan menggunakan algoritma genetika. Masukan aktifitas dan kondisi atlet akan menjadi inputan untuk proses kondisi dan kebutuhan gizi. Juga masukan parameter genetika yaitu *crossover*, mutasi dan

maksimum generasi yang akan digunakan untuk proses optimasi menggunakan algoritma genetika, hasil perhitungan tersebut berupa macam-macam komposisi makanan hasil dari optimasi dengan algoritma genetika. Output inilah yang akan menjadi acuan user untuk menjadi solusi pada permasalahan optimasi komposisi bahan pangan harian atlet yang memenuhi kebutuhan gizi.

3.2.1 Perancangan Diagram Alir

Implementasi algoritma genetika pada persoalan ini secara umum terdiri dari beberapa tahapan.



Gambar 3.3 Diagram sederhana algoritma genetika

Dari diagram alir diatas dapat dijelaskan proses pertama yaitu diawali dengan tahap pembangkitan populasi awal secara acak. Populasi ini terdiri dari kromosom-kromosom. Populasi merupakan kumpulan beberapa kromosom. Proses selanjutnya yaitu proses kedua adalah proses seleksi. Metode seleksi yang

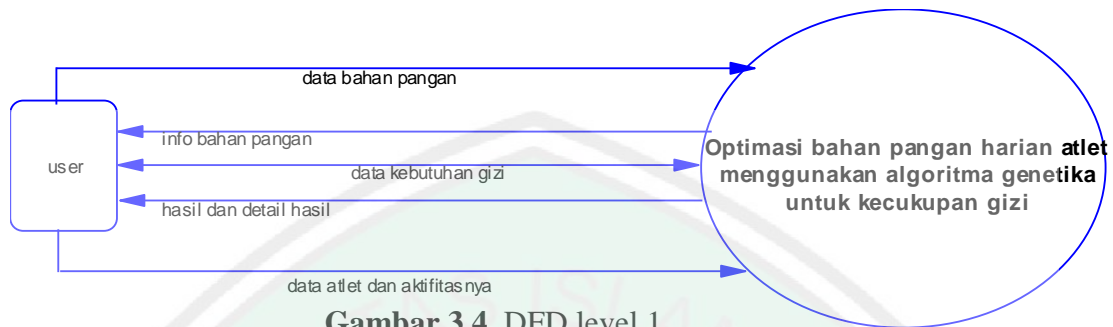
dipilih adalah roda roulette, dan akan dilanjutkan dengan langkah ketiga yaitu langkah *crossover*. *Crossover* adalah suatu proses untuk menyilangkan dua buah kromosom sehingga didapat kromosom yang mempunyai karakteristik yang mirip dengan induknya. Proses *crossover* dilakukan agar kromosom yang dianggap tidak pantas untuk dipertahankan pada generasi selanjutnya bisa digantikan dengan kromosom yang lebih baik dari induknya. Teknik *crossover* yang digunakan adalah *crossover* dua titik. Kemudian langkah selanjutnya adalah langkah keempat proses mutasi. Proses mutasi merupakan proses dimana suatu gen akan mengalami penyimpangan dari kromosom induknya sehingga sifat kromosom anak tersebut akan mengalami perbedaan dari kromosom induknya. Banyaknya jumlah kromosom yang akan dimutasi tergantung dari probabilitas yang telah ditentukan nilainya. Setelah menyelesaikan semua proses proses terakhir adalah proses optimasi, disini akan ditentukan apakah optimasi tercapai atau tidak. Apabila optimasi sudah tercapai maka proses akan selesai namun apabila sebaliknya proses akan kembali pada proses seleksi dan dilanjutkan dengan proses berikutnya sampai mendapatkan hasil yang maksimal untuk menyelesaikan permasalahan optimasi penentuan bahan pangan harian atlet sehingga mencapai kecukupan gizi.

3.2.2 Data Flow Diagram (DFD)

DFD ini menjelaskan proses yang ada di Progam Sistem Pendukung

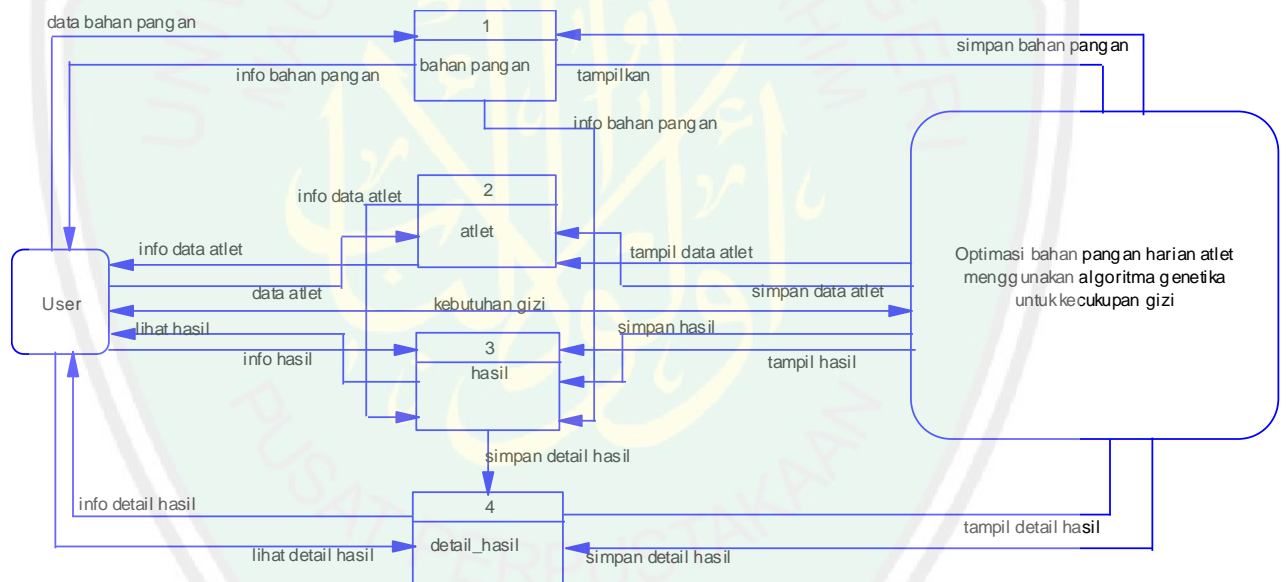
Keputusan optimasi komposisi bahan pangan harian atlet

a. DFD level 1



Gambar 3.4 DFD level 1

b. DFD level 2



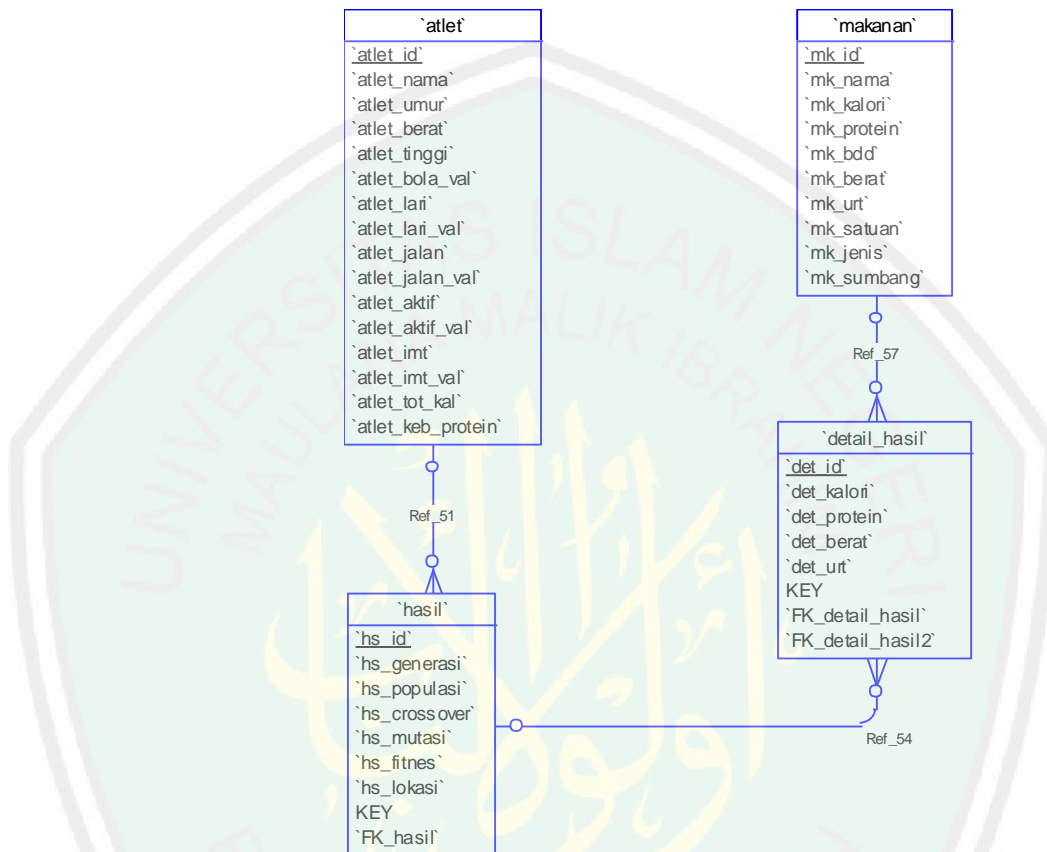
Gambar 3.5 DFD level 2

3.2.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan hubungan tabel yang digunakan dalam table. Di dalam ERD ditampilkan struktur tabel secara keseluruhan. ERD dibagi menjadi dua bagian, yaitu *Conceptual Data Model (CDM)* dan *Physical Data Model (PDM)*.

Berikut gambar dari CDM dan PDM.

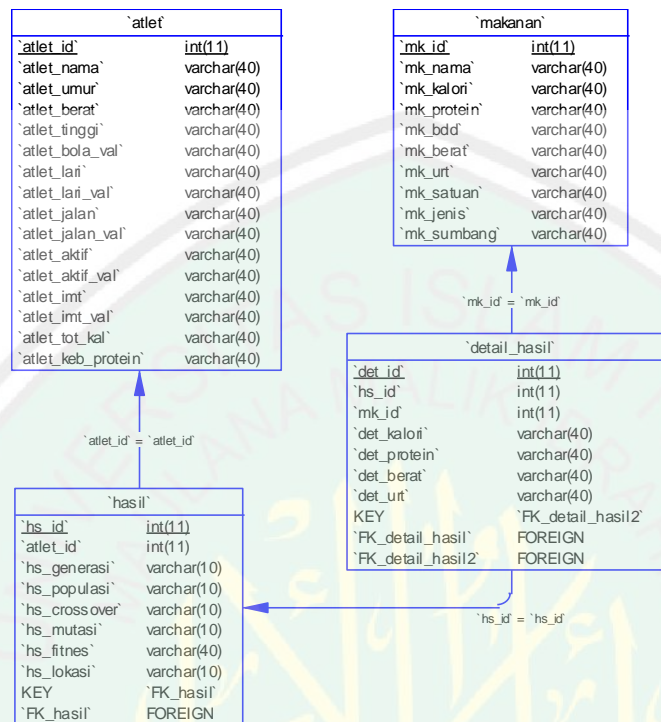
a. CDM



Gambar 3.6 CDM

Tabel-tabel yang digunakan dalam proses optimasi komposisi bahan pangan harian atlet sepakbola dengan metode algoritma genetika (*genetic algorithm*) adalah tabel makanan, atlet, hasil dan detail_hasil. Tabel makanan berisi data makan dan kandungan gizi dalam setiap jenis makanan untuk digunakan system untuk proses genetika kelak. Tabel atlet berisi data aktifitas atlet. Tabel hasil berisi hasil makanan tersimpan dari proses genetika, dan hasil detailnya disimpan dalam tabel detail_hasil.

b. PDM



Gambar 3.7 PDM

3.2.4 Struktur Tabel

Struktur tabel merupakan penjabaran dan penjelasan dari fungsi masing-masing tabel yang digunakan dalam sistem. Adapun struktur tabel adalah sebagai berikut :

1. Table makanan

Tabel makanan berfungsi untuk menampung data makanan dan kandungan gizinya guna proses genetika.

Tabel 3.10 Tabel makanan

Kolom	Tipe Data	Panjang	Keterangan		
			PK	FK	Tabel
mk_id	Int	11	√		
mk_nama	Varchar	40			
mk_kalori	Varchar	40			
mk_protein	Varchar	40			
mk_bdd	Varchar	40			
mk_berat	Varchar	40			
mk_urt	Varchar	40			
mk_satuan	Varchar	40			
mk_jenis	Varchar	40			
mk_sumbang	Varchar	40			

2. Tabel Atlet

Tabel atlet menampung data atlet dan aktifitas atlet.

Tabel 3.11 Tabel atlet

Kolom	Tipe Data	Panjang	Keterangan		
			PK	FK	Tabel
atlet_id	Int	11	√		
atlet_nama	Varchar	40			
atlet_umur	Varchar	40			

atlet_berat	Varchar	40			
atlet_tinggi	Varchar	40			
atlet_bola_val	Varchar	40			
atlet_lari	Varchar	40			
atlet_lari_val	Varchar	40			
atlet_jalan	Varchar	40			
atlet_jalan_val	Varchar	40			
atlet_aktif	Varchar	40			
atlet_aktif_val	Varchar	40			
atlet_imt	Varchar	40			
atlet_imt_val	Varchar	40			
atlet_tot_kal	Varchar	40			
atlet_keb_protein	Varchar	40			

3. Tabel Hasil

Pada tabel hasil ini menampung dari proses genetika.

Tabel 3.12 Tabel hasil

Kolom	Tipe Data	Panjang	Keterangan		
			PK	FK	Tabel
hs_id	Int	11	√		
atlet_id	Int	11		√	
hs_generasi	Varchar	10			

hs_populasi	Varchar	10			
hs_crossover	Varchar	10			
hs_mutasi	Varchar	10			
hs_fitnes	Varchar	40			
hs_lokasi	Varchar	10			

4. Tabel Detail Hasil

Tabel detail hasil ini menampung data makanan hasil dari prose genetika.

Tabel 3.13 Tabel detail hasil

Kolom	Tipe Data	Panjang	Keterangan		
			PK	FK	Tabel
det_id	Int	11	√		
hs_id	Int	11		√	
mk_id	Int	11		√	
det_kalori	Varchar	40			
det_protein	Varchar	40			
det_berat	Varchar	40			
det_urt	Varchar	40			

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Penerapan program aplikasi Optimasi penentuan bahan pangan harian atlet sepakbola menggunakan algoritma genetika ini untuk menentukan komposisi bahan pangan yang memenuhi kecukupan gizi. Digunakan sebagai media informasi bagi masyarakat umum dan atlet juga menejemen sepakbola khususnya untuk mengambil keputusan dalam menentukan komposisi makanan bagi atlet. Dengan adanya program ini diharapkan gizi yang dibutuhkan atlet dapat tercukupi guna menunjang kemampuan maksimal atlet dalam latihan keseharian dan juga dalam melakukan pertandingan.

4.2 Penjelasan Program

Di sini akan dijelaskan alur pembuatan dan kegunaan program yang disertai tampilan desain. Berikut ini beberapa tampilan form yang ada dalam program:

4.2.1 Form Awal

Form Awal adalah menu utama merupakan gerbang masuk ke dalam program optimasi ini. Pada form ini, disediakan beberapa menu diantaranya menu, data dan bantuan. Adapun desain form-nya adalah sebagai berikut:



Gambar 4.1 Tampilan form pertama *software*

Form awal diatas sekaligus sebagai interface pertama yang akan menyapa pengguna atau user. Penulis berusaha menampilkan Gambar yang dapat mewakili aplikasi ini. Form awal ini terdapat tiga tombol pilihan di kiri atas yaitu menu, data dan bantuan. Pada tombol menu terdapat tiga pilihan lagi yaitu home yang berfungsi untuk kembali ke form awal lagi, yang kedua tombol proses untuk melakukan proses genetika tentunya dengan input data atlet terlebih dulu di dalamnya. Pilihan ketiga adalah keluar, dengan memilih tombol keluar berarti user akan keluar dari aplikasi ini, menutup aplikasi ini.

Setelah tombol menu yang memiliki tiga fungsi tadi, sebelah menu ada tombol data. Pada tombol data tersebut pengguna dapat melakukan input dan edit data bahan pangan atau makanan yang akan tersimpan pada database. Data makanan tersebut akan menjadi bahan proses optimasi dengan menggunakan algoritma genetika.

4.2.2 Form Data Bahan Pangan

No	Nama	Kalori	Protein	BDD	Berat	URT	Satuan	Jenis	Sumba...
1	Beras giling	360	6.8	100	100	0.75	Gls	1	58.4
2	Beras merah	359	7.5	100	100	0.75	Gls	1	58.4
3	Jagung	361	8.7	100	100	0.75	Gls	1	58.4
4	Ubi jalar	123	1.8	86	150	1	Bj sdg	2	8.4
5	Tepung sagu	353	0.7	100	40	7	Sdm	2	8.4
6	Daging ayam	302	18.2	68	50	1	Ptg sdg	3	6.5
7	Daging sapi	207	18.8	100	50	1	Ptg sdg	3	6.5
8	Telur ayam	162	12.8	90	60	1	Btr bsr	3	6.5
9	Kacang panjang	44	2.7	75	50	2	Ptg bsr	4	5.3
10	Alpokate	85	0.9	61	50	0.5	Bh bsr	5	2.0
11	Apel	58	0.3	68	75	0.5	Bh sdg	5	2.0
12	Susu sapi	61	3.2	100	200	1	Gls	6	7.0
13	Yogurt	52	3.3	100	200	1	Gls	6	7.0
14	Cuka apel	26.4	0	100	0	1	Sdm	7	6.5

Gambar 4.2 Tampilan form bahan pangan

Form berikutnya adalah form makanan, digunakan untuk input data makanan beserta kandungannya yang meliputi nama, kalori, protein, bagian yang dapat dimakan (BDD), berat, URT atau ukuran rumah tangga, satuan, dan jenis. Data tersebut sudah disiapkan pada table bab sebelumnya. Setelah data sdh dimasukan, user tinggal memilih tombol simpan dan akan otomatis tersimpan. Sedangkan tombol hapus berfungsi menghapus data yang sudah tersimpan. Jika hanya ingin sekedar mengedit, user cukup memilih makanan yang sudah tersimpan dan tampil di sisi knan, setelah memilih makanan otomatis data makan akan muncul pada sisi kiri, jadi tinggal mengedit saja. Dan seperti biasa pada sisi pojok kanan atas terdapat tombol exit dan minimize untuk aplikasi ini.

4.2.3 Form Atlet

Pada form utama tadi terdapat tombol menu yang di dalam nya terdapat pilihan tombol proses. Tombol tersebut menuju pada form atlet berikut:

No	Nama	Umur	Berat	Tinggi	Menit	Latha	Menit	Latha	Menit	Aktivitas	IMT	Kebut	Kebut
1	Ahmad	20	60	170	60	5,5 m...	30	10 m...	20	Kerja...	Normal	4106...	102.6...

Gambar 4.3 Tampilan form atlet

Pada sisi form kiri ini berfungsi untuk pengisian data dari atlet. Mulai dari nama, umur, berat dan tinggi. Juga untuk proses perhitungan kebutuhan Kalori dan protein maka user diharuskan melakukan proses input latihan lari yaitu kecepatannya dan lama berlari, demikian pula latihan jalan. Setelah data terisi dan pilih simpan maka data akan tersimpan dalam *database* dan ditampilkan pada sisi kanan form ini. Sama seperti form sebelumnya, jika *user* ingin melakukan proses *edit*, maka cukup dengan memilih atlet yang akan di *edit*, maka data atlet tersebut akan muncul pada sisi kiri dan *user* bisa melakukan proses edit.

Pada form ini terdapat tombol proses, yang berfungsi sebagai eksekusi perhitungan kebutuhan kalori dan protein dari atlet yang sudah dimasukan datanya tadi. Jangan lupa untuk memilih atlet yang akan diproses baru tekan proses. Hasil *eksekusi* tersebut akan dimunculkan pada pada form berikutnya yaitu form optimasi.

4.2.4 Form Optimasi

Setelah memilih atlet yang datanya sudah tersimpan pada proses sebelumnya di form atlet dan juga form baha pangan sudah terisi maka user akan masuk pada form optimasi:



Gambar 4.4 Tampilan form optimasi

Pada form ini setelah diketahui kebutuha kalori dan protein dari salah satu atlet yang ditampilkan diatas, maka *user* memasukan parameter yang dibutuhkan untuk proses genetika seperti generasi, populasi, *crossover*, mutasi. Aktifitas genetika dari atlet tersebut akan muncul pada sisi kanan table. Juga akan muncul hasil dari proses tersebut dengan memilih tombol hasil akhir. Dilanjutkan disisi pling kanan akan menampilkan hasil akhir yang sudah tersimpan.

4.3 Deskripsi Sistem

Lima proses utama dalam program ini, yaitu inisialisasi populasi, proses hitung fitness, proses seleksi induk, proses perkawinan silang (*crossover*), dan proses mutasi. Pada setiap proses akan dijelaskan sebagai berikut:

4.3.1 Inisialisasi populasi

Inisialisasi populasi dalam program adalah membangkitkan nilai random data makanan sebanyak jumlah makanan yang akan diproses sebagai makanan yang terpilih. Dalam program ini sesuai dengan data kelompok bahan makan dan hasil terbaik pada penelitian sebelumnya akan ditentukan jumlah bahan pakan atau gen sebanyak 9 dan jumlah kromosom sebanyak 20, minimal kromosom dalam algoritma genetika adalah 30. Berikut adalah cuplikan program untuk inisialisasi kromosom:

```
private void viewData() {
    try {
        ResultSet rs = kon.getData("select * from
makanan order by mk_id");
        rs.last();
        int n = rs.getRow();
        kodeMK = new Vector();
        Object data [][] = new Object[n][10];
        int i = 0;
        int j = 1;
        rs.beforeFirst();

        while(rs.next()){
            kodeMK.addElement(rs.getString(1));
            data[i][0] = j++;
            data[i][1] = rs.getString(2);
            data[i][2] = rs.getString(3);
            data[i][3] = rs.getString(4);
            data[i][4] = rs.getString(5);
            data[i][5] = rs.getString(6);
            data[i][6] = rs.getString(7);
            data[i][7] = rs.getString(8);
            data[i][8] = rs.getString(9);
        }
    }
}
```

```

        data[i][9] = rs.getString(10);

        //index = i+1;
        i++;
    }

    jTable1.setModel(new
    javax.swing.table.DefaultTableModel(
        data,
        new String [] {
            "No", "Nama", "Kalori", "Protein", "BDD",
            "Berat", "URT", "Satuan", "Jenis", "Sumbang"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false, false,
            false, false, false, false, false
        };

        public boolean isCellEditable(int rowIndex,
        int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    jScrollPane1.setViewportView(jTable1);

    jTable1.getColumnModel().getColumn(0).setMaxWidth(30);
    jTable1.getColumnModel().getColumn(1).setMinWidth(150);
    jTable1.getColumnModel().getColumn(8).setMinWidth(100);
    } catch (SQLException ex) {

    Logger.getLogger(makanan.class.getName()).log(Level.SEVERE,
    null, ex);
    }
}

```

4.3.2 Hitung Fitness

```

private double hitungFitnes(kromosom c){
    double kalori = 0, protein = 0, bdd = 0, sumbang =
    0, berat = 0, urt = 0, hsKalori = 0, hsProtein = 0, hsBerat
    = 0, hsURT = 0;
    double totKalori = 0, totProtein = 0;

    for(int i=0; i<GenSize; i++){
        kalori = Double.parseDouble(c.kalori[i]);
        protein = Double.parseDouble(c.protein[i]);
        bdd = Double.parseDouble(c.bdd[i]);
        sumbang = Double.parseDouble(c.sumbang[i]);
    }
}

```

```

berat = Double.parseDouble(c.berat[i]);
urt = Double.parseDouble(c.urt[i]);

hsKalori = (sumbang*kebKalori)/100;
hsBerat = (hsKalori/kalori)*100*(100/bdd);
hsProtein = (hsBerat/100)*(bdd/100)*protein;
hsURT = (hsBerat/berat)*urt;

c.hsKalori[i] = String.valueOf(hsKalori);
c.hsProtein[i] = String.valueOf(hsProtein);
c.hsBerat[i] = String.valueOf(hsBerat);
c.hsURT[i] = String.valueOf(hsURT);

totKalori += hsKalori;
totProtein += hsProtein;
}

c.totKalori = String.valueOf(totKalori);
c.totProtein = String.valueOf(totProtein);

double    fitnes    =    1/(Math.abs(kebKalori-
totKalori)+Math.abs(kebProtein-totProtein))+1;

return fitnes;
}

```

Fungsi evaluasi dimaksudkan supaya mendapatkan nilai fitness dari suatu kromosom. Bdd di atas adalah bagian makanan yang dapat dimakan, sumbang adalah sumbangan / presentase terhadap komposisi, urt adalah ukuran dalam rumah tangga, hsUrt adalah hasil ukuran rumah tangga.

4.3.3 Seleksi

Seleksi dimulai dengan dibangkitkan sebuah nilai secara random atau dengan mengacak 1 – sebanyak kromosom dan didapat randTemp(jumlah kromosom yang akan dipilih), kemudian mengambil randTemp kromosom dari jumlah kromosom dan di simpan sebagai Ptemp(populasi temporary). Kemudian mengambil kromosom dengan fitness tertinggi dari randTemp kromosom pada populasi temporary, misalkan didapat kromosom ke-8. Maka parent 1 adalah

kromosom ke-8 begitu juga parent 2 melakukan proses seleksi seperti parent 1 akan tetapi jika mendapatkan hasil yang sama dengan parent 1 maka harus mengulang proses seleksi tersebut sehingga mendapatkan hasil yang berbeda, misalkan didapat kromosom ke-5. Maka parent 2 adalah kromosom ke-5. Berikut cuplikan program untuk memilih induk:

```
private kromosom seleksi(){
    int randTemp;
    do{
        randTemp = (int) (Math.random() *
PopulationSize);
    }
    while(randTemp == 0);

    Vector idTemp = new Vector();
    kromosom[] kromTemp = new kromosom[randTemp];
    for(int i=0; i<kromTemp.length; i++){
        int randID;
        do{
            randID = (int) (Math.random() *
PopulationSize);
        }
        while(idTemp.contains(randID));
        kromTemp[i] = Indv[randID];
        idTemp.addElement(randID);
    }
    return sorting(kromTemp, kromTemp.length);
}
```

4.3.4 Crossover

```

private kromosom crossover(kromosom p1, kromosom p2){
    int k,l,m;
    kromosom c=new kromosom(GenSize);
    int cut = (int)(Math.random()*(GenSize/2));
    int cut2;
    do{
        cut2 = (int) (Math.random() * GenSize);
    }
    while(cut2 <= cut);

    pros.jTextArea1.append("\n Cut point pada posisi
"+(cut+1)+" dan "+(cut2+1));

    for (k = 0; k <= cut; k++){
        c.idMak[k] = p1.idMak[k];
        c.nama[k] = p1.nama[k];
        c.kalori[k] = p1.kalori[k];
        c.protein[k] = p1.protein[k];
        c.bdd[k] = p1.bdd[k];
        c.berat[k] =p1.berat[k];
        c.urt[k] = p1.urt[k];
        c.satuan[k] = p1.satuan[k];
        c.jenis[k] = p1.jenis[k];
        c.sumbang[k] = p1.sumbang[k];
    }

    for (l = cut + 1; l <= cut2; l++){
        c.idMak[l] = p2.idMak[l];
        c.nama[l] = p2.nama[l];
        c.kalori[l] = p2.kalori[l];
        c.protein[l] = p2.protein[l];
        c.bdd[l] = p2.bdd[l];
        c.berat[l] =p2.berat[l];
        c.urt[l] = p2.urt[l];
        c.satuan[l] = p2.satuan[l];
        c.jenis[l] = p2.jenis[l];
        c.sumbang[l] = p2.sumbang[l];
    }

    for (m = cut2 + 1; m < GenSize; m++){
        c.idMak[m] = p1.idMak[m];
        c.nama[m] = p1.nama[m];
        c.kalori[m] = p1.kalori[m];
        c.protein[m] = p1.protein[m];
        c.bdd[m] = p1.bdd[m];
        c.berat[m] =p1.berat[m];
        c.urt[m] = p1.urt[m];
        c.satuan[m] = p1.satuan[m];
        c.jenis[m] = p1.jenis[m];
        c.sumbang[m] = p1.sumbang[m];
    }
}

```


Setelah proses seleksi dilakukan, kemudian nilai random dari proses pemilihan kromosom kandidat orang tua tersebut diperbandingkan dengan peluang *crossover* (P_c), jika nilai random tersebut kurang dari P_c maka kromosom kedua orang tua yang telah dipilih tersebut berpeluang untuk dilakukan proses *crossover* yang kemudian hasilnya disimpan dalam kromosom anak, jika tidak demikian maka kromosom anak adalah kromosom orang tua itu sendiri. Pada perkawinan silang dimulai dengan menyimpan kromosom orang tua pada sebuah variabel. Setelah itu ditentukan titik potong perkawinan silang secara *random*. Dalam masalah ini perkawinan silang diimplementasikan dengan skema *order crossover*, di mana dalam skema ini satu bagian kromosom dipertukarkan dengan tetap menjaga nilai gen dan susunan yang bukan bagian dari kromosom tersebut.

4.3.5 Mutasi

Pada operator mutasi ini digunakan cara *swapping mutation*, yaitu dengan menukar nilai dua gen. Langkah awal dalam mutasi ini menggunakan metode penukaran gen, jadi penukaran 1 gen dengan bahan makanan yang tidak ada didalam gen dengan cara mengambil secara acak jika telah didapat gen yang baru kemudian hitung kembali fitnessnya, kemudian acak mutasi dikurangi 1. Berikut adalah *listing* program untuk operator mutasi:

```
private kromosom mutasi(kromosom c){
    int randGen = (int) (Math.random() * GenSize);
    String idMak = c.idMak[randGen];
    String idMak_baru;
    makanan mak;
    do{
        int randMak = (int) (Math.random() *
makanan[randGen].size());
```

```

        mak = (algoritma.makanan)
makanan[randGen].elementAt(randMak);
        idMak_baru = mak.idMak;
    }
    while(idMak_baru.equals(idMak));

    c.idMak[randGen] = mak.idMak;
    c.nama[randGen] = mak.nama;
    c.kalori[randGen] = mak.kalori;
    c.protein[randGen] = mak.protein;
    c.bdd[randGen] = mak.bdd;
    c.berat[randGen] = mak.berat;
    c.urt[randGen] = mak.urt;
    c.satuan[randGen] = mak.satuan;
    c.jenis[randGen] = mak.jenis;
    c.sumbang[randGen] = mak.sumbang;

    return c;
}

```

4.3.6 Hasil Akhir

Metod dibawah ini untuk menampilkan kromosom dengan fitness terbaik dari generasi yang terpilih, hasil nama makanan, kalori, protein, urt, berat, dan satuan.

```

private void hasilAkhir(){
    Object[][] data = new Object[GenSize][7];
    int j = 0;
    for(int i=0; i<GenSize; i++){
        data[j][0] = j+1;
        data[j][1] = finals.nama[i];
        data[j][2] = finals.hsKalori[i];
        data[j][3] = finals.hsProtein[i];
        data[j][4] = finals.hsBerat[i];
        data[j][5] = finals.hsURT[i];
        data[j][6] = finals.satuan[i];

        j++;
    }

    double bandKalori =
roundTwoDecimals(Double.parseDouble(finals.totKalori)
kebKalori) * 100;
    double bandProtein =
roundTwoDecimals(Double.parseDouble(finals.totProtein)
kebProtein) * 100;

```

```

pros.jTextField3.setText(String.valueOf(bandKalori)+" %");
pros.jTextField4.setText(String.valueOf(bandProtein)+" %");

        pros.jTable1.setModel(new      DefaultTableModel(data,
new String [] {
            "No", "Nama", "Kalori", "Protein", "Berat",
"URT", "Satuan"
        }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false,
false
            };

            public boolean isCellEditable(int rowIndex, int
columnIndex) {
                return canEdit [columnIndex];
            }
        });
        pros.jScrollPane2.setViewportViewView(pros.jTable1);
pros.jTable1.getColumnModel().getColumn(0).setMaxWidth(30);
pros.jTable1.getColumnModel().getColumn(1).setMinWidth(100);
    }

```

4.3.7 Simpan hasil

Untuk menyimpan hasil diatas sebagai berikut:

```

public void simpanHasil(){
    try {
        int id = 1;
        String sql = "select max(hs_id) from hasil";
        ResultSet rs = kon.getData(sql);
        if(rs.next()){
            id += rs.getInt(1);
        }

        sql = "insert into hasil values ('"+id+"',
'"+idAtlet+"', '"+GenerationSize+"', '"+PopulationSize+"',
'"+CrossOverRate+"',           '"+MutationRate+"',
'"+finals.fitness+"', '"+finals.generasi+"')";
        kon.executeUpdate(sql);

        for(int i=0; i<finals.idMak.length; i++){
            sql = "insert into detail_hasil(hs_id,
mk_id, det_kalori, det_protein, det_berat, det_urt)
values('"+id+"',           '"+finals.idMak[i]+"',

```

```

"+finals.hsKalori[i]+"',          '"+finals.hsProtein[i]+"',
 '"+finals.hsBerat[i]+"', '"+finals.hsURT[i]+"'");
        kon.setUpdate(sql);
    }

    viewDataDB();
    JOptionPane.showMessageDialog(null, "Data telah
tersimpan.", "Informasi", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {

Logger.getLogger(genetika.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
}

```

4.4 Uji Coba Aplikasi

Setelah disapa oleh tampilan awal uji coba dimulai dengan memasukan data dan aktifitas atlet pada tampilan input atlet. Tampilan input atlet terdapat dalam menu pilihan proses.

The screenshot shows a software window titled "OPTIMASI GIZI ATLET SEPAK BOLA DENGAN METODE ALGORITMA GENETIKA". On the left, there is an "Input Atlet" form with the following fields:

- Nama: cakmil
- Umur: 30 thn
- Berat: 60 kg
- Tinggi: 170 cm
- Latihan Sepak Bola: 30 menit
- Latihan Lari: 5,5 menit/km (dropdown) and 5 menit (input)
- Latihan Jalan: 10 menit/km (dropdown) and 10 menit (input)
- Tingkat Aktivitas: Kerja ringan (dropdown)

 Below the form are "Simpan" and "Hapus" buttons. On the right, a table displays the following data:

No	Nama	Umur	Berat	Tinggi	Menit	Lathi...	Menit...	Lathi...	Menit...	Aktivitas	IMT	Kebut...	Kebut...
1	Ahmad	20	60	170	60	5,5 m...	30	10 m...	20	Kerja ...	Normal	4106...	102.6...
2	cakmil	30	60	170	30	5,5 m...	5	10 m...	10	Kerja ...	Normal	2927...	73.18...

 Below the table is a "Proses" button.

Gambar 4.5 Tampilan input data atlet

Setelah data dimasukan dan disimpan akan muncul data atlet disebelah kanan tampilan. Sorot salah satu data atlet untuk proses edit atau hapus. Sedangkan pilihan proses untuk menghitung kebutuhan kalori dan protein atlet. Perhitungan kebutuhan kalori dan protein sebagai berikut

```

{
    aktif_val = 1.2;
}else if(aktifm == 1){
    aktif_val = 1.4;
}else if(aktifm == 2){
    aktif_val = 1.5;
}else if(aktifm == 3){
    aktif_val = 1.7;
}else if(aktifm == 4){
    aktif_val = 1.8;
}else if(aktifm == 5){
    aktif_val = 2.1;
}else if(aktifm == 6){
    aktif_val = 2.3;
}

if(nama.length() <= 0 || umur.length() <= 0 || berat.length() <= 0 ||
tinggi.length() <= 0 || sepak.length() <= 0 || lari_val.length() <= 0 || jalan_val.length() <=
0){
    JOptionPane.showMessageDialog(null, "Data tidak boleh kosong",
"Peringatan", JOptionPane.WARNING_MESSAGE);
    return;
}

double imt_val =
Double.parseDouble(berat)/((Double.parseDouble(tinggi)/100)*(Double.parseDouble(tin
ggi)/100));
String imt = "";
if(imt_val < 17){
    imt = "Kurus Sekali";
}else if(imt_val >= 17 && imt_val <= 18.4){
    imt = "Kurus";
}else if(imt_val > 18.4 && imt_val <= 25){
    imt = "Normal";
}else if(imt_val > 25 && imt_val <= 27){

```

```

        imt = "Gemuk";
    }else if(imt_val > 27){
        imt = "Gemuk Sekali";
    }

    double bmr = 0;
    if(berat.equals("55") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1514;
    }else if(berat.equals("55") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1499;
    }else if(berat.equals("60") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1589;
    }else if(berat.equals("60") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1556;
    }else if(berat.equals("65") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1664;
    }else if(berat.equals("65") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1613;
    }else if(berat.equals("70") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1739;
    }else if(berat.equals("70") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1670;
    }else if(berat.equals("75") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1814;
    }else if(berat.equals("75") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){

```

```

        bmr = 1727;
    }else if(berat.equals("80") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1889;
    }else if(berat.equals("80") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1785;
    }else if(berat.equals("85") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 1964;
    }else if(berat.equals("85") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1842;
    }else if(berat.equals("90") && (Integer.parseInt(umur) >= 18 &&
Integer.parseInt(umur) < 30)){
        bmr = 2039;
    }else if(berat.equals("90") && (Integer.parseInt(umur) >= 30 &&
Integer.parseInt(umur) <= 60)){
        bmr = 1899;
    }

    double sda = 0.1*bmr;
    double tot_kal = bmr+sda;
    tot_kal = aktif_val*tot_kal;

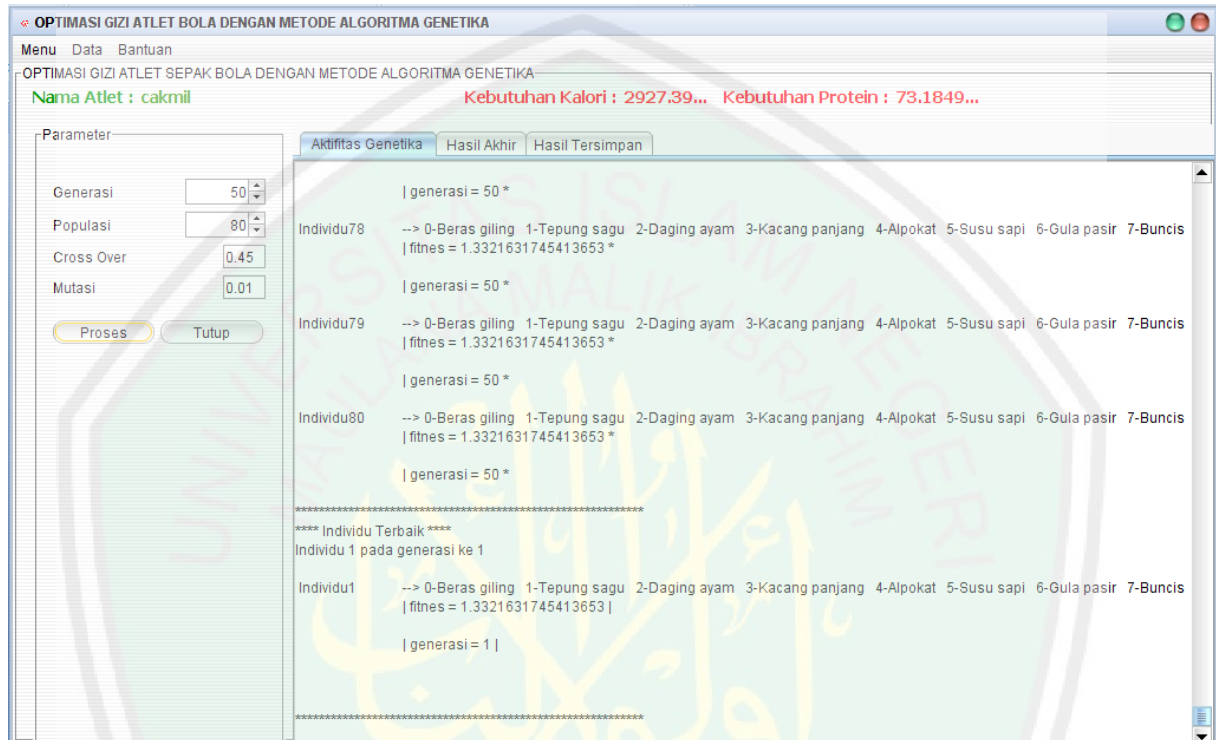
    tot_kal = tot_kal+kal_bola+kal_lari+kal_jalan;

    if(imt.equals("Kurus Sekali")){
        tot_kal += 500;
    }else if(imt.equals("Gemuk Sekali")){
        tot_kal -= 500;
    }

    double keb_protein = tot_kal*0.1*0.25;
    String sql = "";

```

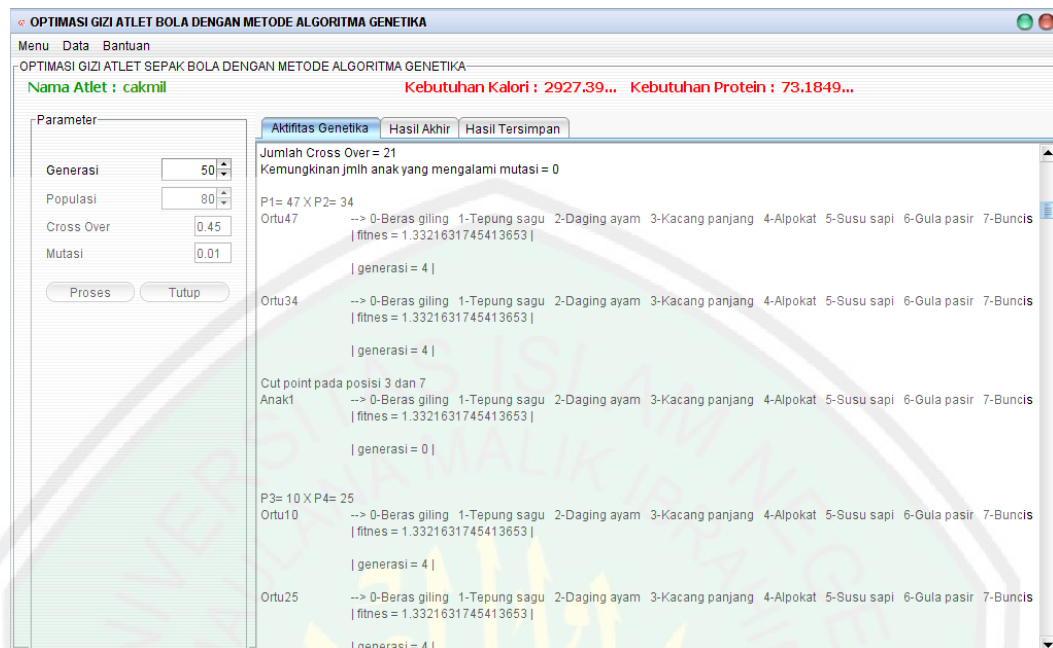
Dan setelah tombol proses dipilih maka akan muncul hasil kebutuhan kalori dan protein pada tampilan dibawah ini



Gambar 4.6 Tampilan proses genetika

Kebutuhan kalori dan protein sudah didapat yaitu 2927, 39 dan 73,1849. Untuk memulai aktifitas genetika kita masukan parameter nya dengan generari 50, populasi 80, probabilitas *crossover* 0,45 dan probabilitas mutasinya 0,01. Hasil aktifitas genetikan muncul pada tampilan sebelah kanan.

Namun sebelum keluar hasil terdapat proses *crossover* dan mutasi, berikut cuplikan prosesnya



Gambar 4.7 Tampilan proses *crossover*

Jumlah Cross Over = 21
 Kemungkinan jmlh anak yang mengalami mutasi = 0

P1= 66 X P2= 70
 Ortu66 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
 | fitness = 1.3321631745413653 |
 | generasi = 36 |

Ortu70 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
 | fitness = 1.3321631745413653 |
 | generasi = 36 |

Cut point pada posisi 4 dan 9
 Anak1 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
 | fitness = 1.3321631745413653 |

| generasi = 0 |

P3= 27 X P4= 4

Ortu27 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 |

| generasi = 36 |

Ortu4 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 |

| generasi = 36 |

Cut point pada posisi 3 dan 4

Anak2 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 |

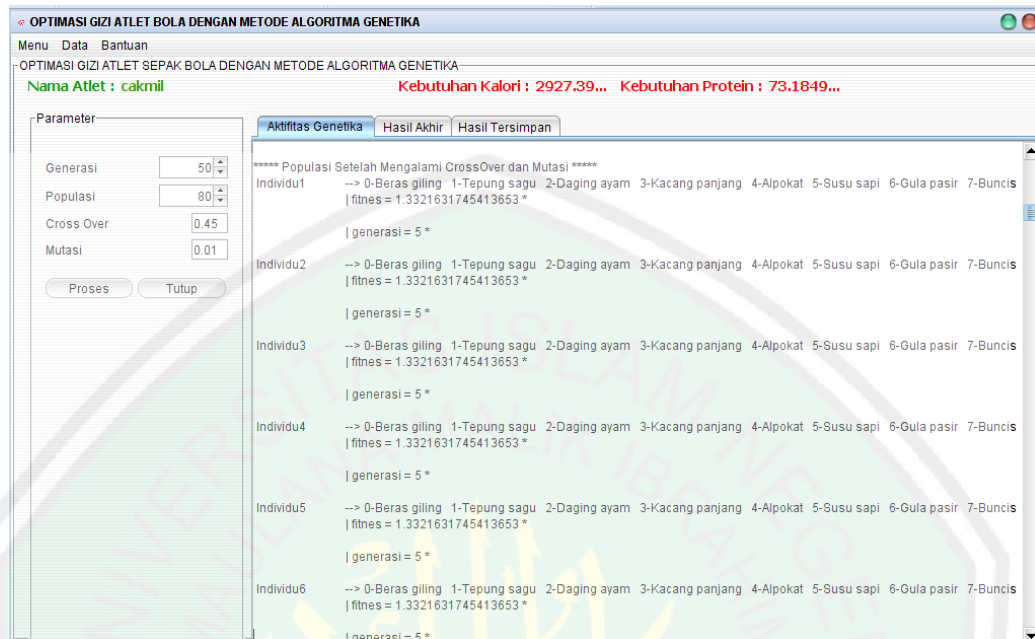
| generasi = 0 |

P5= 6 X P6= 46

Ortu6 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 |

| generasi = 36 |

Ortu46 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang
4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 |



Gambar 4.8 Proses mutasi

```

***** Populasi Setelah Mengalami CrossOver dan Mutasi *****
Individu1    --> 0-Beras giling  1-Tepung sagu  2-Daging ayam  3-Kacang
panjang  4-Alpokat  5-Susu sapi  6-Gula pasir  7-Buncis  8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu2    --> 0-Beras giling  1-Tepung sagu  2-Daging ayam  3-Kacang
panjang  4-Alpokat  5-Susu sapi  6-Gula pasir  7-Buncis  8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu3    --> 0-Beras giling  1-Tepung sagu  2-Daging ayam  3-Kacang
panjang  4-Alpokat  5-Susu sapi  6-Gula pasir  7-Buncis  8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

```

Individu4 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu5 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu6 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu7 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu8 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

| generasi = 5 *

Individu9 --> 0-Beras giling 1-Tepung sagu 2-Daging ayam 3-Kacang panjang 4-Alpokat 5-Susu sapi 6-Gula pasir 7-Buncis 8-Minyak kelapa
| fitness = 1.3321631745413653 *

Dan muncul hasil akhirnya pada menu hasil akhir seperti berikut

OPTIMASI GIZI ATLET BOLA DENGAN METODE ALGORITMA GENETIKA

Menu Data Bantuan

OPTIMASI GIZI ATLET SEPAK BOLA DENGAN METODE ALGORITMA GENETIKA

Nama Atlet : cakmil Kebutuhan Kalori : 2927.39... Kebutuhan Protein : 73.1849...

Parameter:

Generasi: 50
Populasi: 80
Cross Over: 0.45
Mutasi: 0.01

Simpan

No	Nama	Kalori	Protein	Berat	URT	Satuan
1	Beras giling	2398.03248000000...	45.2961690666666...	666.12013333333...	4.9959010000000...	Gls
2	Tepung sagu	344.92248000000...	0.6839822549575...	97.711750708215...	17.099556373937...	Sdm
3	Daging ayam	266.9043	16.084961125827...	152.37742635304...	3.0475485270609...	Ptg sdg
4	Kacang panjang	217.62966	13.354547318181...	659.48381818181...	26.379352727272...	Ptg bsr
5	Apel	82.124400000000...	0.4247813793103...	208.22616632860...	1.3881744421906...	Bh sdg
6	Susu sapi	287.4354	15.078578360655...	471.20557377049...	2.3560278688524...	Gls
7	Gula pasir	217.62966	0.0	59.788368131868...	7.4735460164835...	sdm
8	Buncis	164.2488000000...	11.262774857142...	521.424761904762...	5.2142476190476...	Gls
9	Minyak kelapa	123.1866	0.1415937931034...	14.159379310344...	1.4159379310344...	sdm

Perbandingan:

Kalori = 100.0 %
Protein = 100.0 %

Gambar 4.9 Tampilan hasil akhir

4.5 Optimasi Dalam Pandangan Islam

Salah satu usaha manusia untuk mendapatkan hasil yang terbaik dari pilihan yang tersedia adalah tindakan optimasi. Dengan begitu manusia berusaha untuk merubah keadaan menjadi lebih baik. Tanpa disadari manusia melakukan optimasi diberbagai kegiatannya. Benar benar selaras dengan firman Allah SWT :

لَهُ مُعَقِّبَاتٌ مِّنْ بَيْنِ يَدَيْهِ وَمِنْ خَلْفِهِ يَحْفَظُونَهُ مِنْ أَمْرِ اللَّهِ إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّىٰ يُغَيِّرُوا مَا بِأَنْفُسِهِمْ وَإِذَا أَرَادَ اللَّهُ بِقَوْمٍ سُوءًا فَلَا مَرَدَّ لَهُ وَمَا لَهُمْ مِّنْ دُونِهِ مِنِّ وَالٍ

Artinya :

bagi manusia ada malaikat-malaikat yang selalu mengikutinya bergiliran, di muka dan di belakangnya, mereka menjaganya atas perintah Allah. Sesungguhnya Allah tidak merubah keadaan sesuatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri. dan apabila Allah menghendaki keburukan terhadap sesuatu kaum, Maka tak ada yang dapat menolaknya; dan sekali-kali tak ada pelindung bagi mereka selain Dia.(QS. Ar-Ra'd : 11)

Ayat di atas menjelaskan bagi tiap-tiap manusia ada beberapa Malaikat yang tetap menjaganya secara bergiliran dan ada pula beberapa Malaikat yang mencatat amalan-amalannya. dan yang dikehendaki dalam ayat ini ialah Malaikat yang menjaga secara bergiliran itu, disebut Malaikat Hafazhah. Tuhan tidak akan merubah keadaan mereka, selama mereka tidak merubah sebab-sebab kemunduran mereka.

Dalam ayat tersebut Alloh SWT mensyaratkan manusia untuk selalu berusaha untuk mendapatkan yang terbaik dalam kehidupan. Optimasi disini sejalan dengan hal tersebut, dimana manusia senantiasa berusaha mendapatkan yang terbaik. Dengan melakukan teknik optimasi dalam berbagai permasalahan.

Alloh SWT pun tidak menyukai hambanya yang melakukan pemborosan. Dengan begitu optimasi sangat dibutuhkan untuk pemecahan yang berdasar pada keefisienan menjauhi pemborosan. Alloh SWT berfirman :

إِنَّ الْمُبَذِّرِينَ كَانُوا إِخْوَانَ الشَّيْطَانِ ۗ وَكَانَ الشَّيْطَانُ لِرَبِّهِ كَفُورًا

Artinya :

Sesungguhnya pemboros-pemboros itu adalah saudara-saudara syaitan dan syaitan itu adalah sangat ingkar kepada Tuhannya. (QS. Al-Israa' : 27)

Dengan begitu jelas Alloh SWT melarang hambaNya bersifat boros yang menyerupai tindakan syetan yang telah ingkar pada Alloh SWT. Dengan proses optimasi diharapkan mendapatkan hasil yang baik dari beberapa pilihan yang ada. Dan menghindari sifat pemborosan.

BAB V

PENUTUP

5.1 Kesimpulan

Telah terbukti penerapan algoritma genetika dapat diterapkan pada optimasi penentuan bahan pangan harian atlet sepakbola. Optimasi dengan algoritma genetika dapat membantu optimalisasi penentuan bahan pangan yang memenuhi kecukupan gizi bagi atlet. Namun begitu aplikasi ini masih belum menemukan optimasi yang maksimum, atau belum tercapai komposisi bahan pangan yang sempurna. Selain kesempurnaan memang hanya milik Allah SWT pencipta alam semesta, juga memang aplikasi ini masih terdapat kelemahan seperti penentuan golongan makanan yang terbaru. Juga takaran yang spesifik untuk makanan bagi atlet tersebut.

Penggolongan aktifitas dari atlet sepakbola juga belum spesifik. Karena itu akan berpengaruh pada perhitungan kebutuhan kalori yang tentunya berbeda dari setiap atlet yang melakukan latihan berbeda. Kebutuhan kalori yang akurat membuat proses eksekusi genetika menghasilkan hasil yang optimal. Tak lupa juga penggolongan input data dari atlet, dimana begitu beragamnya data yang dimiliki atlet seperti berat badan, tinggi, umur dan lainnya yang beragam satu dengan yang lainnya.

5.2 Saran

Dengan kekurangan yang ada pada aplikasi ini penulis menyarankan beberapa hal untuk dijadikan pengembangan guna mendapatkan aplikasi yang lebih baik :

1. Perlu diadakan penelitian secara langsung untuk mengetahui seberapa besar pengaruh hasil dari optimasi aplikasi ini terhadap perkembangan Atlet.
2. Perlu adanya penelitian lebih lanjut guna pengembangan aplikasi ini. Seperti halnya penggolongan bahan makanan dan penggolongan jenis aktifitas atlet sepakbola. Mungkin juga tambahan output menu yang bisa disajikan pada atlet.

DAFTAR PUSTAKA

Al-Qur'an Al-Karim

Almatsier, Sunita. 2003. *Prinsip Dasar ILMU GIZI*. Gramedia Pustaka Utama, Jakarta.

Desiani, Anita dan Muhammad Arhami. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Andi.a

Direktorat Gizi Masyarakat, Ditjen Binkesmas, Depkes RI, 2002.

Pedoman Umum Gizi Seimbang, Jakarta

Kusumadewi, Sri. 2007. *Aplikasi Teknologi Informasi*. SNATI 2007. Yogyakarta.

Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu, Yogyakarta.

Kusumadewi, Sri dan Hari Purnomo. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu.

Kuswadi, Son. 2007. *Kendali Cerdas, Teori dan Aplikasi Praktisnya*. Yogyakarta: Andi.

Suyanto. 2006. *Algoritma Genetika dalam MATLAB*. Andi, Jogjakarta

Suyanto. 2007. *Artificial Intelligence: Searching, Reasoning, Planning and Learning*. Bandung: Informatika.

Tafsir Ibnu Katsir

Yusuf, Liswarti. 2008. *Teknik Perencanaan Gizi Makanan Jilid 2*. Depdiknas, Jakarta