

**OPTIMASI KOMPOSISI PAKAN SAPI POTONG  
MENGUNAKAN ALGORITMA GENETIKA**

**SKRIPSI**

**LABORATORIUM KOMPUTASI DAN SISTEM CERDAS**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer



Disusun oleh  
**JASICKA INDRI KUSUMA**  
NIM. 115060801111063

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI  
UNIVERSITAS BRAWIJAYA**

**PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

**PROGRAM STUDI INFORMATIKA/ ILMU KOMPUTER**

**MALANG**

**2015**

## LEMBAR PERSETUJUAN

### OPTIMASI KOMPOSISI PAKAN SAPI POTONG MENGUNAKAN ALGORITMA GENETIKA

### SKRIPSI

### LABORATORIUM KOMPUTASI DAN SISTEM CERDAS

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer



Disusun oleh:

**JASICKA INDRI KUSUMA**

**NIM. 115060801111063**

Skripsi ini telah diperiksa dan disetujui oleh dosen pembimbing

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Wayan Firdaus Mahmudy, S.Si., MT., Ph.D**

**NIP. 19720919 199702 1 001**

**Indriati, S.T., M.Kom**

**NIK. 831013 06 1 2 0035**

**LEMBAR PENGESAHAN**

**OPTIMASI KOMPOSISI PAKAN SAPI POTONG  
MENGUNAKAN ALGORITMA GENETIKA**

**SKRIPSI**

Disusun oleh:

**JASICKA INDRI KUSUMA**

**NIM. 115060801111063**

Skripsi ini telah diuji dan dinyatakan lulus  
tanggal 29 Mei 2015

**Dosen Penguji I**

**Dosen Penguji II**

**Dian Eka Ratnawati, S.Si., M.Kom**  
**NIP. 19730619 200212 2 001**

**M. Tanzil Furqon, S.Kom, M.CompSc**  
**NIP. 19820930 20801 1 004**

**Dosen Penguji III**

**Wijaya Kurniawan, S.T., M.T**  
**NIK. 820125 16 1 1 0418**

Mengetahui,  
Ketua Program Studi Informatika/Ilmu Komputer

**Drs. Marji, M.T**  
**NIP. 19670801 199203 1 001**

## PERNYATAAN ORISINALITAS SKRIPSI

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

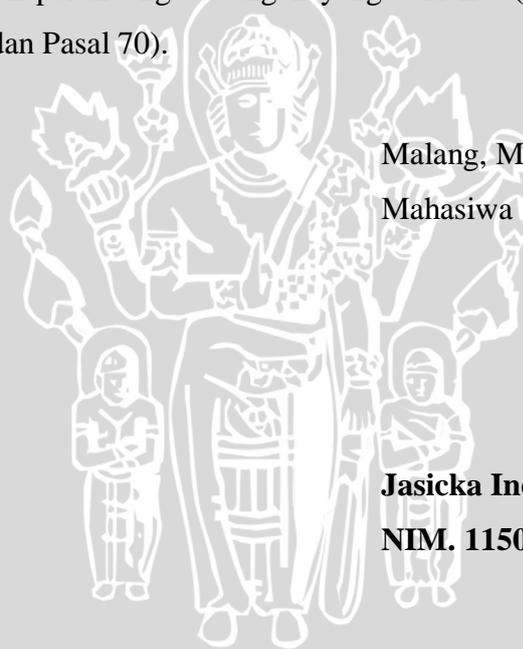
Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, Mei 2015

Mahasiwa

**Jasicka Indri Kusuma**

**NIM. 115060801111063**



## KATA PENGANTAR

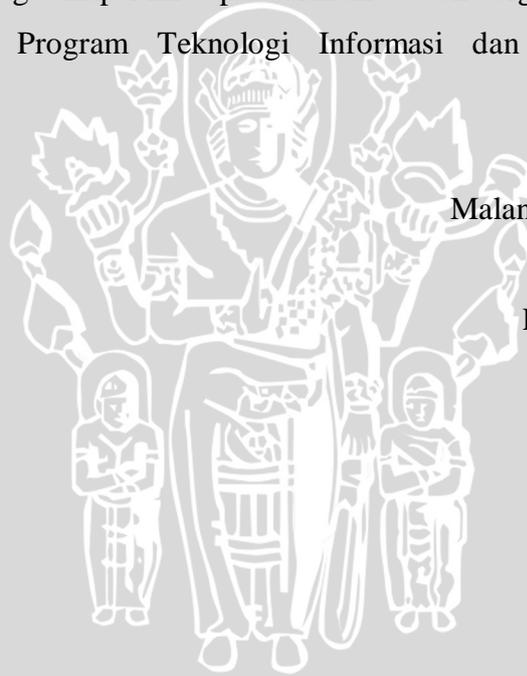
Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Penyayang. Puji syukur penulis panjatkan kehadirat Allah SWT karena dengan rahmat dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**Optimasi Pakan Sapi Potong Menggunakan Algoritma Genetika**”. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika/Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penyusunan skripsi ini juga tak lepas dari bantuan semua pihak yang terus memberikan semangat, kritik, saran, bimbingan, serta doa. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T.,Ph.D. selaku dosen pembimbing I yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Ibu Indriati, S.T., M.Kom. selaku dosen pembimbing II yang telah banyak memberikan ilmu dan saran untuk laporan skripsi ini.
3. Bapak Aswin Suharsono, S.T., M.T. selaku dosen pembimbing akademik yang telah banyak memberikan saran dan motivasi selama masa perkuliahan.
4. Drs. Marji. M.T. dan Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Informatika/Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Informatika/Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Seluruh Civitas Akademika Informatika/Ilmu Komputer Universitas Brawijaya yang telah banyak memberi dukungan dan bantuan selama penulis menempuh studi di Informatika/Ilmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
7. Kedua orang tua penulis, Bapak Suryanto dan Ibu Subandiyah yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil. Adik penulis Rizaldi Arydian Febriyanto dan seluruh keluarga besar yang telah memberikan semangat dari awal sampai akhir pengerjaan skripsi ini.

8. Teman-teman penulis Ladies, Himyatul, Meidina, Arini, Yasmin, Syifa, seluruh teman-teman IF kelas I, serta teman-teman angkatan 2011, angkatan 2012, dan angkatan 2013 yang selalu memberi dukungan, motivasi, kritik, dan saran.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.



Malang, Mei 2015

Penulis

## ABSTRAK

**Jasicka Indri Kusuma. 2015: Optimasi Komposisi Pakan Sapi Potong Menggunakan Algoritma Genetika. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Wayan Firdaus Mahmudy, S.Si., MT., Ph.D dan Indriati, S.T., M.Kom.**

Daging memiliki manfaat yang sangat besar untuk pemenuhan gizi berupa protein hewani. Produktivitas ternak sapi potong sebagai salah satu sumber makanan berupa daging masih memprihatinkan karena volume produksi daging yang masih rendah. Salah satu usaha perbaikan produksi daging sapi potong adalah penyediaan pakan yang dapat memenuhi kebutuhan nutrisi ternak. Namun, biaya pembelian ransum ternak merupakan biaya tertinggi dalam usaha peternakan. Agar memperoleh biaya minimal, penelitian ini menawarkan solusi menggunakan algoritma genetika sebagai algoritma untuk penentuan komposisi pakan sapi potong.

Data yang digunakan dalam penelitian ini adalah 31 data bahan pakan sapi beserta kandungan nutrisinya. Proses algoritma genetika ini menggunakan representasi *real-code* dengan panjang kromosom sesuai dengan bahan pakan yang tersedia, metode crossover yang digunakan adalah *extended intermediate crossover*, metode mutasi yang digunakan adalah *random mutation*, dan diseleksi dengan metode *elitism*. Dari hasil pengujian yang dilakukan diperoleh parameter optimal yaitu ukuran populasi sebesar 200 individu dengan rata-rata fitness sebesar 0.4411528, generasi sebanyak 100 dengan rata-rata fitness sebesar 0.441953 dan kombinasi *cr* 0.5 dan *mr* 0.1 dengan rata-rata fitness sebesar 0.4400628. Hasil akhir berupa komposisi pakan sapi harian yang memenuhi kebutuhan nutrisi dengan meminimumkan biaya pakan.

**Kata kunci:** Algoritma Genetika, Sapi Potong, Komposisi Pakan

## ABSTRACT

**Jasicka Indri Kusuma. 2015: Optimization Food Composition of Beef Cattle with Genetic Algorithms. Program of Information Technology and Computer Science, Brawijaya University, Malang. Advisor: Wayan Firdaus Mahmudy, S.Si., MT., Ph.D and Indriati, S.T., M.Kom.**

Beef has a huge benefit of nutrition as animal protein. The productivity of cattle as a source of food like beef is still a concern because of the beef production volume is still low. One of efforts to increase beef cattle production is supplying feed that can fullfil the nutrition needs of beef cattle. However, the cost of purchasing cattle is the highest costs in the livestock business. In order to obtain a minimal cost, this research offers solutions using genetic algorithm as the algorithm for determination of the composition of the feed beef cattle.

The data used in this research is the 31 data of beef cattle feed along with the nutritional content. The process of genetic algorithms using real-code representation with the length of the chromosome depending to the available feed, the crossover method used is extended intermediate crossover, mutation method used random mutaion, and selected methods of elitism. The test results obtained optimal parameters such as the size of the population of 200 individuals with average fitness of 0.4411528, generation of 100 with average fitness of 0.441953 and the combination of cr 0.5 and mr 0.1 with average fitness of 0.4400628. The end result is a composition of daily cattle feed that fullfil nutritional needs by minimizing the cost of the feed.

**Keywords:** Genetic Algorithms, Beef Cattle, Food Composition

**DAFTAR ISI**

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR PERSAMAAN.....	xiii
DAFTAR <i>SOURCE CODE</i> .....	xiv
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
<b>BAB II.....</b>	<b>6</b>
<b>KAJIAN PUSTAKA DAN DASAR TEORI.....</b>	<b>6</b>
2.1 Kajian Pustaka.....	6
2.2 Sapi Potong.....	8
2.3 Bahan Pakan Sapi Potong.....	9
2.4 Kebutuhan Nutrisi Sapi Potong.....	12
2.5 Ransum.....	17
2.6 Algoritma Genetika.....	19
2.7 Parameter algoritma genetika.....	21
2.8 Penerapan Algoritma Genetika.....	22
2.8.1 Representasi Kromosom.....	22
2.8.2 Inisialisasi.....	23
2.8.3 Reproduksi.....	23
2.8.4 Evaluasi.....	26



2.8.5	Seleksi.....	26
BAB III.....		31
METODOLOGI PENELITIAN .....		31
3.1	Tahapan Penelitian.....	31
3.2	Kebutuhan Sistem.....	32
3.3	Formulasi Permasalahan .....	32
3.4	Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika .....	34
3.4.1	Representasi Kromosom dan Perhitungan Fitness .....	36
3.4.2	Reproduksi .....	41
3.4.3	Evaluasi dan Seleksi .....	45
BAB IV .....		48
PERANCANGAN SISTEM.....		48
4.1	Perancangan <i>Database</i> .....	48
4.2	Perancangan Antarmuka Aplikasi .....	49
4.2.1	Rancangan Halaman Utama.....	49
4.2.2	Rancangan Halaman Bahan Pakan.....	50
4.2.3	Rancangan Halaman Parameter.....	51
4.3	Perancangan Uji Coba dan Evaluasi.....	52
4.3.1	Uji Coba Ukuran Populasi .....	52
4.3.2	Uji Coba Banyaknya generasi .....	53
4.3.3	Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i> .....	53
BAB V.....		55
IMPLEMENTASI SISTEM.....		55
5.1	Implementasi Program.....	55
5.1.1	Implementasi Proses untuk Inisialisasi Populasi Awal.....	55
5.1.2	Implementasi Proses untuk Perhitungan <i>Fitness</i> .....	57
5.1.3	Implementasi Proses untuk <i>Crossover</i> .....	58
5.1.4	Implementasi Proses untuk Mutasi.....	60
5.1.5	Implementasi Proses untuk Seleksi .....	62
5.2	Implementasi Antarmuka Aplikasi .....	63
5.2.1	Tampilan Halaman Utama .....	63
5.2.2	Tampilan Halaman Bahan Pakan .....	66
5.2.3	Tampilan Halaman Parameter .....	66

BAB VI .....	67
PENGUJIAN DAN ANALISIS .....	67
6.1    Sistematika Pengujian .....	67
6.2    Analisis dan Pembahasan .....	67
6.2.1    Pengujian dan Analisis Ukuran Populasi .....	68
6.2.2    Pengujian dan Analisis Banyaknya Generasi .....	69
6.2.3    Pengujian dan Analisis Kombinasi <i>Cr</i> dan <i>Mr</i> .....	71
6.3    Solusi Terbaik yang Pernah Didapatkan .....	73
BAB VII .....	74
PENUTUP .....	74
7.1    Kesimpulan .....	74
7.2    Saran .....	74
DAFTAR PUSTAKA .....	75
LAMPIRAN .....	77



## DAFTAR GAMBAR

Gambar 2.1 <i>Pseudocode</i> Algoritma Genetika .....	20
Gambar 2.2 Flowchart Algoritma Genetika .....	21
Gambar 2.3 <i>Pseudocode</i> Seleksi <i>Elitism</i> .....	27
Gambar 3.1 Diagram Blok Tahap Penelitian .....	31
Gambar 3.2 <i>Flowchart</i> Perhitungan Ransum dengan Algoritma Genetika .....	35
Gambar 3.3 <i>Flowchart</i> Subproses Pembentukan Populasi Awal .....	35
Gambar 3.4 Flowchart Perhitungan <i>Fitness</i> .....	39
Gambar 3.5 Flowchart Proses <i>Crossover</i> .....	42
Gambar 3.6 Flowchart Proses Mutasi .....	44
Gambar 3.7 Flowchart Proses Seleksi <i>Elitism</i> .....	46
Gambar 5.1 Halaman Utama .....	64
Gambar 5.2 Detail Perhitungan .....	64
Gambar 5.3 Kromosom Terbaik .....	65
Gambar 5.4 Kromosom Terbaik .....	65
Gambar 5.5 Halaman Bahan Pakan .....	66
Gambar 5.6 Halaman Parameter .....	66
Gambar 6.1 Grafik Hasil Uji Coba Ukuran Populasi .....	70
Gambar 6.2 Grafik Hasil Uji Coba Banyaknya generasi .....	71
Gambar 6.3 Grafik Hasil Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i> .....	73

## DAFTAR TABEL

Tabel 2.1 Perbedaan Penelitian Sebelumnya dengan Skripsi Penulis .....	7
Tabel 2.2 Populasi Sapi Potong di Indonesia .....	8
Tabel 2.3 Hasil Analisis Kimia Beberapa Bahan Pakan Konsentrat .....	10
Tabel 2.4 Kebutuhan Gizi Sapi Jantan untuk Pertumbuhan dan Penggemukan ...	14
Tabel 2.5 Kebutuhan Gizi Sapi Betina untuk Pertumbuhan dan Penggemukan ...	15
Tabel 2.6 Kemampuan Sapi Mengonsumsi Pakan .....	16
Tabel 2.7 Contoh Representasi Kromosom.....	23
Tabel 2.8 Hasil <i>Offspring</i> .....	26
Tabel 2.9 Kumpulan Individu.....	28
Tabel 2.10 Individu Hasil Seleksi Elitism.....	28
Tabel 3.1 Populasi Awal .....	36
Tabel 3.2 Nilai PK dan TDN Masing-Masing Bahan.....	37
Tabel 3.3 Kebutuhan dan Ketersediaan PK dan TDN.....	38
Tabel 3.4 Nilai <i>Fitness</i> .....	40
Tabel 3.5 Perhitungan Manual <i>Crossover</i> .....	43
Tabel 3.6 Perhitungan Manual Mutasi.....	44
Tabel 3.7 Hasil Perhitungan <i>Fitness</i> Seluruh Individu.....	45
Tabel 3.8 Hasil Seleksi <i>Elitism</i> .....	46
Tabel 3.9 Pemilihan Kromosom Terbaik.....	47
Tabel 4.1 Pemilihan Kromosom Terbaik .....	52
Tabel 4.2 Rancangan Uji Coba Banyaknya Generasi.....	53
Tabel 4.3 Rancangan Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i> .....	54
Tabel 6.1 Hasil Uji Coba Ukuran Populasi .....	68
Tabel 6.2 Hasil Uji Coba Banyaknya generasi.....	70
Tabel 6.3 Hasil Uji Coba Kombinasi <i>Cr</i> dan <i>Mr</i> .....	72

## DAFTAR PERSAMAAN

Persamaan (2-1) Rumus <i>Schoorl</i> .....	17
Persamaan (2-2) Rumus <i>Lambourne</i> .....	17
Persamaan (2-3) Rumus <i>Offspring Hasil Crossover</i> .....	26
Persamaan (2-4) Rumus Gen Hasil Mutasi.....	26
Persamaan (2-5) Rumus <i>Fitness</i> Pencarian Nilai Maksimum.....	26
Persamaan (2-6) Rumus <i>Fitness</i> Pencarian Nilai Minimum.....	26
Persamaan (2-7) Rumus Total <i>Fitness</i> .....	29
Persamaan (2-8) Rumus Probabilitas Seleksi.....	29
Persamaan (2-9) Rumus Probabilitas Kumulatif.....	29
Persamaan (3-1) Rumus Kandungan Nutrisi.....	37
Persamaan (3-2) Rumus Harga.....	38
Persamaan (3-3) Rumus <i>Fitness</i> .....	37



**DAFTAR SOURCE CODE**

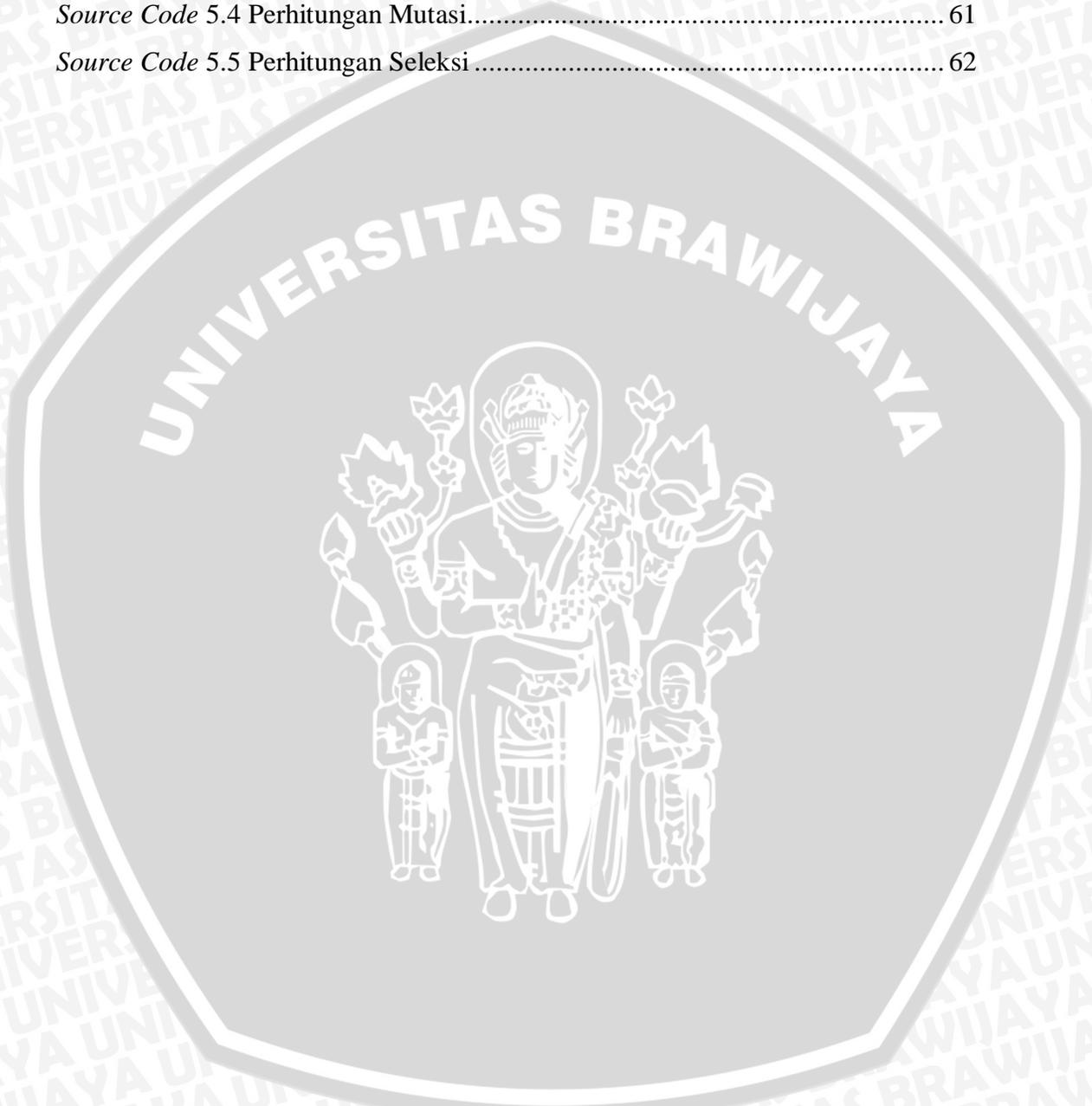
*Source Code 5.1* Inisialisasi Populasi Awal..... 56

*Source Code 5.2* Perhitungan *Fitness* ..... 58

*Source Code 5.3* Perhitungan *Crossover* ..... 60

*Source Code 5.4* Perhitungan Mutasi..... 61

*Source Code 5.5* Perhitungan Seleksi ..... 62



## BAB I

### PENDAHULUAN

Pada bab ini akan dibahas beberapa hal, yaitu latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

#### 1.1 Latar Belakang

Ternak sapi potong merupakan salah satu sumber daya penghasil daging yang memiliki ekonomi tinggi. Daging memiliki manfaat yang sangat besar untuk pemenuhan gizi berupa protein hewani. Pada tahun 1983 LIPI merekomendasikan bahwa masyarakat Indonesia rata-rata memerlukan 50 gram protein per harinya. Namun, kebutuhan protein masyarakat Indonesia hanya 20 % yang berasal dari protein hewani dan sisanya berasal dari protein nabati. Rendahnya konsumsi protein hewani di Indonesia membuat para peternak perlu untuk meningkatkan produksi daging, seperti daging sapi potong (Sudarmono, 2008).

Produktivitas ternak sapi potong sebagai salah satu sumber makanan berupa daging masih memprihatinkan karena volume produksi daging yang masih rendah. Salah satu faktor yang menyebabkan volume produksi daging yang rendah adalah terbatasnya bahan pakan yang tersedia (Sudarmono, 2008). Maka dari itu, perlu adanya penyusunan ransum yang efisien agar produksi daging dapat meningkat.

Salah satu usaha perbaikan produksi daging sapi potong adalah penyediaan pakan yang dapat memenuhi kebutuhan nutrisi ternak (Sudarmono, 2008). Namun, dalam sudut pandang ekonomi, biaya pembelian ransum ternak merupakan biaya tertinggi dalam usaha peternakan. Biaya tersebut harus ditekan serendah mungkin untuk memaksimalkan pendapatan (Nugraha, 2011).

Dalam penelitian sebelumnya yang dilakukan oleh Wardhani, Safrizal dan Chairin pada tahun 2011 tentang optimasi komposisi bahan pakan ikan air tawar menggunakan metode algoritma genetika multi objective. Tujuan yang ingin dicapai adalah meminimalkan perbedaan gizi pada komposisi bahan yang dihasilkan sistem dengan kebutuhan nilai gizi optimalnya. Inisialisasi populasi yang digunakan adalah

metode integer encoding dengan pemenuhan kebutuhan nutrisi mencapai 100% dan efisiensi biaya sekitar 46,5%. Penelitian lainnya dilakukan pada tahun 2008 oleh Ariwibowo, Lukas, dan Gunawan menggunakan metode algoritma genetika untuk menentukan komposisi pakan ayam petelur dengan tujuan untuk mendapatkan komposisi pakan terbaik dalam rangka meningkatkan produksi dan kualitas ayam.

Terdapat berbagai metode optimasi yang dapat digunakan untuk menyelesaikan permasalahan penyusunan komposisi bahan pakan, diantaranya adalah *Simulated Annealing (SA)*, *Tabu Search (TS)*, *Particle Swarm Optimization (PSO)*, *Evaluation Strategies (ES)*, *Genetic Algorithm (GA)* dan lain sebagainya. Metode algoritma genetika telah berhasil diterapkan pada berbagai permasalahan yang kompleks, maka dari itu penulis akan menggunakan algoritma genetika sebagai algoritma untuk penentuan komposisi pakan sapi potong. Proses pencarian solusi diawali dengan pembangkitan solusi secara acak. Populasi awal terdiri dari kromosom-kromosom dimana setiap kromosom tersebut merepresentasikan komposisi bahan pakan yang digunakan untuk mendapatkan hasil yang optimal. Populasi yang telah dipilih akan menghasilkan keturunan baru yang sifatnya diharapkan lebih baik dari populasi sebelumnya. Populasi yang baik akan memiliki peluang untuk terus dikembangkan agar menghasilkan keturunan populasi yang lebih baik. Dengan demikian, solusi terbaik yang diinginkan dapat diperoleh dengan mengulang proses pencarian keturunan (Aribowo, 2008).

Dengan adanya sistem ini diharapkan dapat membantu peternak untuk mengoptimalkan penyusunan komposisi pakan sapi potong yang dapat memenuhi kebutuhan nutrisi dengan biaya yang minimum. Hal ini diharapkan dapat membantu para peternak untuk mengurangi biaya produksi dan meningkatkan hasil produksi daging sapi potong.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah diatas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana mengimplementasikan algoritma genetika untuk menentukan komposisi bahan pakan sapi potong?
2. Bagaimana menentukan parameter algoritma genetika yang tepat?
3. Bagaimana mengukur kualitas solusi yang dihasilkan algoritma genetika?

## 1.3 Batasan Masalah

Batasan masalah yang akan dijadikan sebagai pedoman dalam pelaksanaan penelitian ini adalah sebagai berikut:

1. Penentuan komposisi pakan sapi potong didasarkan pada jenis sapi potong, bobot badan sapi potong, target penambahan bobot badan sapi potong dan bahan pakan yang tersedia.
2. Kebutuhan nutrisi sapi potong yang digunakan adalah protein kasar (PK) dan energi / TDN.
3. Bobot badan sapi potong dibatasi sebanyak 7 macam yaitu 100 kg, 150 kg, 200 kg, 250 kg, 300 kg, 350 kg, dan 400 kg.
4. Pertambahan bobot badan sapi potong dibatasi sebanyak 3 macam yaitu 0,50 kg, 0,75 kg, dan 1,00 kg.
5. Terdapat 31 macam bahan pakan sapi yang akan dioptimasi.
6. Bahan pakan yang akan dioptimasi hanya bahan pakan yang mengandung protein kasar (PK) dan energi / TDN.
7. Sistem rekomendasi ini menghitung pakan tiap satu sapi dalam satu hari dan belum dibandingkan secara langsung ke lapangan.

## 1.4 Tujuan

Tujuan dari penyusunan skripsi ini adalah membangun sistem rekomendasi komposisi pakan sapi potong agar memenuhi kebutuhan nutrisi dengan biaya pakan yang minimal menggunakan algoritma genetika.

### 1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Membantu peternak sapi potong untuk menentukan komposisi bahan pakan sapi potong yang optimal
2. Membantu peternak sapi potong untuk meminimalkan biaya pakan dengan memenuhi kebutuhan nutrisi.

### 1.6 Sistematika Penulisan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir, secara garis besar meliputi beberapa bab sebagai berikut:

- BAB I : Pendahuluan**  
Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.
- BAB II : Kajian Pustaka dan Dasar Teori**  
Menguraikan tentang dasar teori dan referensi yang mendasari pembuatan sistem optimasi komposisi pakan sapi potong menggunakan algoritma genetika.
- BAB III : Metodologi Penelitian**  
Menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari tahapan penelitian, kebutuhan sistem, formulasi permasalahan, dan siklus penyelesaian masalah menggunakan algoritma genetika yang mendasari metodologi pembuatan sistem ini.

**BAB IV : Perancangan Sistem**

Menguraikan tentang perancangan sistem yang terdiri dari perancangan database, perancangan antarmuka aplikasi, dan perancangan uji coba dan evaluasi yang mendasari perancangan pembuatan sistem optimasi komposisi pakan sapi potong menggunakan algoritma genetika.

**BAB V : Implementasi**

Membahas implementasi program dan antarmuka aplikasi dari sistem optimasi komposisi pakan sapi potong menggunakan algoritma genetika yang sesuai dengan perancangan sistem yang telah dibuat.

**BAB VI : Pengujian dan Analisis**

Memuat hasil pengujian dan analisis terhadap sistem rekomendasi yang telah direalisasikan.

**BAB VII : Penutup**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.

## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dibahas beberapa hal, yaitu kajian pustaka, pengertian sapi potong, bahan pakan sapi potong, kebutuhan nutrisi sapi potong, ransum, algoritma genetika, parameter algoritma genetika, dan penerapan algoritma genetika.

#### 2.1 Kajian Pustaka

Terdapat beberapa penelitian tentang komposisi bahan pakan yang telah dilakukan dengan berbagai macam objek dan metode. Skripsi ini menggunakan referensi beberapa penelitian komposisi bahan pakan. Referensi pertama adalah penelitian oleh Wardhani, Safrizal, dan Chairi (2011) dengan judul “Optimasi komposisi Bahan Pakan Ikan Air Tawar Menggunakan Metode Multi-Objective Genetic Algorithm”. Tujuan penelitian tersebut adalah untuk meminimalkan perbedaan gizi pada komposisi bahan yang dihasilkan sistem dengan kebutuhan nilai gizi optimalnya dengan pemenuhan kebutuhan nutrisi mencapai 100% dan efisiensi biaya sekitar 46,5%. Kombinasi optimal untuk menghasilkan pakan sejumlah 6 kg diperoleh dengan parameter panjang gen 5, jumlah kromosom 200, probabilitas crossover 0.01, probabilitas mutasi 0.2, probabilitas elitism 0.03, dan banyaknya generasi 5.

Referensi yang kedua adalah penelitian dengan judul “Penerapan Algoritma Genetika pada Penentuan Komposisi Pakan Ayam Petelur” yang dilakukan oleh Ariwibowo, Lukas, dan Gunawan (2008) dengan tujuan untuk mendapatkan komposisi pakan terbaik dalam rangka meningkatkan produksi dan kualitas ayam. Berdasarkan pengujian yang telah dilakukan dengan parameter algoritma genetika yang bervariasi, hasil yang optimum diperoleh pada pengujian dengan jumlah gen 20, jumlah kromosom 1000, probabilitas crossover 0.5, probabilitas mutasi 0.1, dan banyaknya generasi 4000 dengan nilai fitness 0.92.

Persamaan dan perbedaan dari kedua penelitian dengan skripsi yang diajukan berdasarkan studi literatur yang telah dilakukan terdapat pada Tabel 2.1.

**Tabel 2.1** Perbedaan Penelitian Sebelumnya dengan Skripsi Penulis

No	Judul	Objek	Metode	Perbandingan	
				Studi Literatur	Skripsi Penulis
1	Optimasi komposisi Bahan Pakan Ikan Air Tawar Menggunakan Metode Multi-Objective Genetic Algorithm (Wardhani, 2011)	Ikan Air Tawar	Multi-Objective Genetic Algorithm, Seleksi <i>Roulette Wheel</i>	Ikan Air Tawar	Sapi Potong, Seleksi <i>elitism</i>
2	Penerapan Algoritma Genetika pada Penentuan Komposisi Pakan Ayam Petelur (Ariwibowo, 2008)	Ayam petelur	Algoritma Genetika, Seleksi <i>Roulette Wheel</i>	Ayam Petelur	Sapi Potong, Seleksi <i>elitism</i>

**Sumber:** Diolah oleh penulis

Skripsi ini membahas tentang optimasi komposisi pakan sapi potong menggunakan algoritma genetika. Representasi kromosom yang digunakan adalah representasi *real-code*. Representasi kromosom ini menampilkan representasi berupa angka *real*. Metode *crossover* yang digunakan adalah metode *extended intermediate*, yaitu metode yang menghasilkan *offspring* dari kombinasi nilai dua induk. Sedangkan untuk proses mutasi digunakan metode *random mutation*, yaitu menambah atau mengurangi nilai gen terpilih dengan bilangan *random* yang kecil. Proses yang terakhir merupakan proses seleksi menggunakan metode *elitism*, yang melakukan seleksi pada individu – individu dalam penampungan berdasarkan nilai *fitness* tertinggi.

Beberapa dasar teori yang digunakan pada skripsi ini adalah sapi potong, bahan pakan sapi potong, kebutuhan nutrisi sapi potong, ransum, penyusunan ransum, dan algoritma genetika.

## 2.2 Sapi Potong

Terdapat beberapa jenis sapi potong yang menyebar di wilayah Indonesia seperti Sapi Bali, Madura, Ongole, dan Peranakan ongole (Soeprapto, 2006). Perbandingan populasi sapi-sapi tersebut dapat dilihat pada Tabel 2.2 berikut:

**Tabel 2.2** Populasi Sapi Potong di Indonesia

Jenis Sapi	Jumlah	Presentase
Bali	2.632.125	26,92
Madura	1.131.375	11,57
Ongole	260.094	2,66
Peranakan ongole	773.165	8,17
Lainnya (impor)	4.997.830	50,68

**Sumber:** Soeprapto, 2006

Perbedaan sapi-sapi tersebut akan dijelaskan lebih detail sebagai berikut:

### a. Sapi Bali

Sapi Bali merupakan salah satu jenis sapi asli dari Indonesia yang mempunyai potensi besar untuk dikembangkan. Sapi Bali paling diminati oleh petani-petani kecil di Indonesia karena mempunyai beberapa keunggulan seperti tingkat kesuburan yang tinggi, tipe pekerja yang baik, efisien dalam memanfaatkan sumber pakan, daging rendah lemak, dan daya adaptasi terhadap lingkungan yang tinggi (Soeprapto, 2006).

### b. Sapi Madura

Sapi Madura merupakan sapi lokal Indonesia yang mempunyai potensi plasma nutfah yang perlu dijaga kelestarian dan kemurniannya (Ngadiyono, 2007). Sapi Madura merupakan persilangan antara *Bos sondaicus* dan *Bos indicus* (Sudarmono, 2008).

### c. Sapi Ongole

Sapi ini merupakan sapi potong lokal yang memiliki punuk. Sapi ini berwarna putih dan mempunyai banyak lipatan kulit di bagian leher dan perutnya (Soeprapto, 2006).

d. Sapi Peranakan Ongole

Sapi ini merupakan bangsa sapi yang banyak dipelihara peternak kecil di Pulau Jawa. Sapi ini merupakan persilangan sapi ongole asal India dan sapi Madura (Soeprpto, 2006).

Sapi potong mempunyai klasifikasi yang sama dengan sapi perah. Namun sapi potong adalah sapi yang diperlihara khusus untuk diambil dagingnya. Bangsa sapi mempunyai klasifikasi sebagai berikut (Ngadiyono, 2012):

- Filum : Chordata
- Sub Filum : Vertebrata
- Kelas : Mamalia
- Sub Kelas : Theria
- Ordo : Artiodactyla
- Sub Ordo : Ruminansia
- Famili : Bovidae
- Genus : Bos
- Spesies : Bos Taurus (Sapi Eropa)  
Bos indicus (Sapi India/Sapi Zebu)

### 2.3 Bahan Pakan Sapi Potong

Bahan pakan ternak sapi digolongkan menjadi tiga, yaitu (Sudarmono, 2008):

a. Pakan Hijauan

Pakan hijauan adalah semua bahan pakan yang berasal dari tanaman ataupun tumbuhan berupa daun-daunan. Kelompok pakan hijauan adalah bangsa rumput (Graminae), legum, dan tumbuh-tumbuhan lain. Pakan hijauan tersebut dapat diberikan dalam dua bentuk yaitu pakan hijauan segar dan pakan hijauan kering.

b. Pakan Penguat

Pakan penguat adalah pakan yang berkonsentrasi tinggi dengan kandungan serat kasar yang relatif rendah dan mudah dicerna. Bahan pakan penguat meliputi bahan pakan yang berasal dari biji-bijian seperti jagung giling, menir, bulgur, dedak, katul, bungkil kelapa, tetes, dan berbagai umbi.

c. Pakan Tambahan

Pakan tambahan bagi sapi meliputi vitamin, mineral, urea, dan lain-lain. Vitamin yang dibutuhkan ternak sapi adalah vitamin A dan vitamin D. Sedangkan mineral utama yang dibutuhkan untuk bereproduksi adalah Ca dan P. Urea sebagai bahan pakan tambahan biasanya diberikan dalam jumlah yang terbatas, yaitu 2% dari seluruh ransum yang diberikan.

Hasil analisis kimia bahan pakan konsentrat dapat dilihat pada Tabel 2.3

**Tabel 2.3** Hasil Analisis Kimia Beberapa Bahan Pakan Konsentrat

No	Bahan Pakan	Bahan Kering (%)	Presentase dari Bahan Kering				
			Protein Kasar	Serat Kasar	Lemak Kasar	Beta-N	Energi / TDN
1	Dedak padi kasar	87.5	13.8	8.4	9.4	54.3	5.5
2	Dedak padi halus	89.6	15.9	8.5	9.1	56.7	67.0
3	Dedak terigu kasar	89.3	16.7	9.9	3.5	64.9	-
4	Dedak terigu halus	87.4	18.9	6.9	4.7	65.7	70.0
5	Polard	88.4	17.0	8.8	5.1	45.0	70.0
6	Tepung jagung kuning	89.1	10.8	3.1	4.7	78.9	90.0
7	Tepung gaplek	85.2	2.3	2.8	0.2	78.7	78.0
8	Tepung ikan	89.7	49.0	5.7	4.7	35.5	59.0
9	Tepung terigu	88.2	11.6	1.4	2.8	74.9	-
10	Ampas tahu	26.2	23.7	23.6	10.1	39.0	79.0
11	Ampas bir	85.8	33.7	19.2	6.1	37.1	74.0
12	Ampas kecap	26.6	23.5	16.0	24.2	22.1	87.0
13	Bulgur	90.7	12.9	1.5	1.4	68.5	-
14	Bungkil kelapa	87.9	21.2	13.1	17.3	41.1	81.0
15	Bungkil kelapa sawit	88.6	16.5	15.6	2.5	41.2	70.0
16	Bungkil kacang tanah	80.6	33.7	11.5	13.8	41.9	81.0
17	Tepung biji kapuk	91.0	32.7	16.8	1.7	25.3	74.0

18	Tepung darah	89.2	80.3	5.1	0.8	6.0	-
19	Bungkil kedelai	88.6	41.3	8.6	15.0	21.9	83.2
20	Tetes/molase	87.5	3.1	-	-	85.6	70.7
21	Onggok	88.7	1.8	11.0	0.2	74.1	85.0
22	Dedak merah	86.6	9.6	23.1	6.7	34.6	-
23	Dedak kuning	87.4	9.0	33.2	8.8	22.4	-
24	Dedak jagung	84.8	8.5	1.5	9.0	63.8	82.0
25	Ampas sagu	80.4	1.2	10.8	1.0	83.5	-
26	Bungkil geblek	88.6	23.5	33.9	8.6	15.8	-
27	Bungkil arga	87.8	19.7	23.1	11.2	27.4	-
28	Tepung biji kapas	86.0	36.0	12.0	1.6	28.7	-
29	Kulit buah kakao	88.9	14.6	33.0	11.8	34.9	47.0
30	Kecipir	92.7	39.0	7.3	17.8	31.5	-
31	Bungkil wijen	92.9	42.8	7.0	10.3	26.9	77.3
32	Singkong	32.3	3.3	4.2	3.3	87.7	81.8
33	Ubi jalar	32.0	3.2	3.5	1.4	89.3	83.9
34	Rumput lapangan	21.8	6.7	34.2	1.8	44.2	56.2
35	Rumput serawit	17.9	11.3	23.8	4.8	42.5	-
36	Rumput kampai	31.0	11.5	33.7	2.2	45.7	-
37	Rumput molases	17.2	8.7	33.9	2.1	43.9	53.0
38	Rumput gajah	21.0	9.6	32.7	1.9	45.2	52.4
39	Rumput raja	22.4	13.5	34.1	3.5	30.3	57.0
40	Rumpur setaria	24.0	12.7	35.0	2.0	40.8	54.0
41	Daun jagung	21.0	9.9	27.4	1.8	1.8	-
42	Daun kacang kedelai	22.6	16.7	27.7	3.7	3.7	-
43	Daun kacang tanah	22.8	13.8	25.2	4.9	4.9	-
44	Daun kol luar	9.9	21.5	12.9	3.3	3.3	-
45	Daun lamtoro	24.8	24.2	21.5	3.7	3.7	-
46	Pucuk tebu	25.5	5.2	2.0	33.4	50.2	-
47	Daun turi	28.3	29.2	17.1	3.4	40.1	-
48	Daun singkong	21.6	24.1	4.7	22.1	37.0	61.8
49	Daun pisang	23.3	16.6	23.0	5.2	43.4	73.5
50	Jerami padi	87.5	4.2	32.5	1.5	45.0	43.2
51	Kudzu	32.7	15.1	40.9	1.9	33.6	58.3
52	Batang pisang	7.5	5.9	26.8	2.2	46.9	-

Sumber: Siregar, 2008

## 2.4 Kebutuhan Nutrisi Sapi Potong

Produktivitas ternak sangat dipengaruhi oleh jumlah pakan dan nutrisi yang dapat dimanfaatkan oleh ternak tersebut (Nugroho, 2013). Beberapa bahan makanan mempunyai komposisi yang bervariasi. Adapun beberapa istilah bahan makanan yang perlu diketahui antara lain adalah sebagai berikut (Parakkasi, 1999):

a. Bahan Segar

Bahan segar adalah bahan yang diberikan kepada ternak langsung setelah dipotong / dipanen.

b. Bahan Kering

Diketahui bahwa kadar air bahan makanan bila diberikan kepada ternak (as fed) sangat bervariasi. Oleh karena itu kadar suatu zat dalam suatu bahan makanan akan jauh lebih baik bila dihitung berdasarkan Bahan Kering (BK) .

c. Protein

Tubuh memerlukan protein untuk memperbaiki dan menggantikan sel tubuh yang rusak serta untuk produksi. Protein dalam tubuh diubah menjadi energi jika diperlukan. Protein dapat diperoleh dari bahan-bahan pakan yang berasal dari tumbuh-tumbuhan dan yang berasal dari biji-bijian (Sugeng, 1998 dalam Haryanti, 2009). Protein diperlihatkan dalam tabel komposisi adalah protein kasar (PK).

d. Lemak

Kadar lemak memperlihatkan jumlah lemak kasar (semua zat yang larut dalam eter) yang dikandung oleh bahan makanan.

e. Serat Kasar, ADF dan NDF

Diketahui bahwa serat kasar telah menurun popularitas penggunaannya setelah sekitar 100 tahun digunakan sebagai indikator pencernaan suatu bahan makanan. Acid Detergent Fiber (ADF) adalah serat yang tidak larut dalam deterjen asam. Neutrasi Detergent Fiber (NDF) adalah serat yang tidak larut dalam deterjen netral atau dinding sel.

f. Mineral

Kalsium (Ca) dan Fosfor (P) adalah unsur yang penting untuk diperhatikan pada hampir semua kondisi pemberian pakan. Potasium (K) akan menjadi penting bila level makanan penguat meningkat dalam ransum atau NPN mensubstitusi protein. Belerang (S) juga akan menjadi penting bila NPN dalam ransum meningkat.

g. Energi

Ada empat cara yang dapat digunakan untuk mengukur kadar energi suatu bahan makanan. Walaupun mempunyai beberapa kekurangan, TDN adalah cara/data yang banyak tersedia dan merupakan suatu standar untuk menyatakan nilai energi suatu bahan makanan untuk ruminansia.

Pada penelitian ini, nutrisi yang digunakan untuk memenuhi kebutuhan pakan sapi potong adalah protein kasar (PK) dan energi/TDN. Keseimbangan energi dan protein menjadi hal yang penting karena dapat mempengaruhi dinamika proses fermentasi mikrobial di dalam rumen (Haryanto, 2012).

Protein kasar adalah semua zat yang mengandung nitrogen. Diketahui bahwa dalam protein rata-rata mengandung nitrogen 10%. Secara umum, protein merupakan unsur kritis pada ternak muda, ternak yang tumbuh cepat dan untuk ternak yang berproduksi tinggi. Ternak tidak dapat mengembangkan potensi genetik mereka, tidak dapat menghasilkan tenaga yang maksimal apabila ransum mereka tidak mengandung protein yang cukup (Tim Laboratorium, 2012). Ketersediaan hijauan dan komposisi kimianya terutama protein kasar (PK) adalah faktor pertama yang dipertimbangkan dalam pengembangan program nutrisi yang efektif (Mathis, 2003).

Energi / Total digestible nutrient (TDN) adalah total energi zat makanan pada ternak yang disetarakan dengan energi dari karbohidrat yang dapat diperoleh secara uji biologis ataupun perhitungan menggunakan data hasil analisis proksimat (Tim Laboratorium, 2012). Energi merupakan indikator utama dalam menentukan kebutuhan pakan ruminansia (Haryanto, 2012). Kekurangan energi dapat

mengakibatkan terhambatnya penambahan bobot badan, penurunan bobot badan dan berkurangnya semua fungsi produksi dan terjadi kematian bila berlangsung lama (Tillman et al., 1991 dalam Haryanti, 2009).

Kebutuhan gizi sapi jantan untuk pertumbuhan dan penggemukan dapat dilihat pada Tabel 2.4 dan kebutuhan gizi sapi betina untuk pertumbuhan dan penggemukan dapat dilihat pada Tabel 2.5

**Tabel 2.4** Kebutuhan Gizi Sapi Jantan untuk Pertumbuhan dan Penggemukan

Bobot Badan (kg)	Pertambahan Bobot Badan (kg/hari)	Kebutuhan Zat Gizi				
		Protein Kasar (g)	Energi / TDN (kg)	Ca (g)	P (g)	Vitamin A (1.000 IU)
100	0.5	379	1.6	15	9	6
	0.75	448	1.9	20	11	6
	1	541	2.2	25	15	7
150	0.5	474	2.2	16	10	9
	0.75	589	2.6	21	13	9
	1	607	3	27	16	9
200	0.5	554	2.8	16	12	12
	0.75	622	3.2	21	15	13
	1	690	3.7	27	17	13
	1.1	714	3.9	30	18	13
250	0.5	623	3.2	16	14	13
	0.75	693	3.8	21	17	14
	1	760	4.3	28	19	14
	1.1	782	4.6	30	20	14
300	0.5	679	3.7	19	14	13
	0.75	753	4.3	23	18	15
	1	819	5	28	21	16
	1.1	847	5.3	30	22	16

350	0.50	731	4.1	20	16	18
	0.75	806	4.8	25	18	18
	1.00	874	5.6	30	21	18
	1.10	899	5.9	31	23	18
	1.20	923	6.2	32	24	18
400	0.50	772	4.6	21	18	17
	0.75	875	5.4	26	21	18
	1.00	913	6.2	31	24	19
	1.10	942	6.6	32	25	19
	1.20	967	7.0	33	25	19
	1.30	988	7.2	33	26	19
450	0.50	805	5.0	20	20	17
	0.75	911	5.9	26	23	19
	1.00	952	6.8	29	26	20
	1.10	975	7.2	30	27	20
	1.20	998	7.6	31	28	20
	1.30	1018	7.9	32	29	20

Sumber: Siregar, 2008

**Tabel 2.5** Kebutuhan Gizi Sapi Betina untuk Pertumbuhan dan Penggemukan

Bobot Badan (kg)	Pertambahan Bobot Badan (kg/hari)	Kebutuhan Zat Gizi				
		Protein Kasar (g)	Energi / TDN (kg)	Ca (g)	P (g)	Vitamin A (1.000 IU)
100	0.5	391	1.7	14	11	6
	0.75	460	2	20	14	6
	1	527	2.3	26	18	7
150	0.5	513	2.3	14	12	9
	0.75	552	2.7	19	15	9
	1	623	3.1	25	18	9
200	0.5	577	2.8	14	13	13
	0.75	639	3.3	19	16	13
	1	707	3.8	23	18	15
250	0.5	564	3.3	13	13	14
	0.75	544	3.9	18	15	14
	1	724	4.5	23	18	14
	1.1	757	4.8	25	20	14

300	0.50	604	3.8	14	14	16
	0.75	717	4.5	17	15	16
	1.00	764	5.2	21	18	16
	1.10	797	6.1	24	20	16
350	0.50	637	4.3	15	15	18
	0.75	717	5.0	15	15	18
	1.00	797	5.8	18	18	18
	1.10	829	6.1	20	19	18
	1.20	860	6.4	21	20	18
400	0.50	657	4.7	15	15	19
	0.75	739	5.6	16	16	19
	1.00	819	6.5	18	18	19
	1.10	850	6.8	19	19	19
	1.20	883	7.0	20	19	19

**Sumber:** Siregar, 2008

Kemampuan sapi dalam mengkonsumsi ransum diukur berdasarkan bobot tubuhnya. Semakin besar bobot sapi maka semakin kemampuan mengkonsumsi pakan cenderung semakin menurun. Kemampuan sapi dalam mengkonsumsi pakan dapat dilihat pada Tabel 2.6

**Tabel 2.6** Kemampuan Sapi Mengkonsumsi Pakan

Kisaran bobot badan (kg)	Kemampuan mengkonsumsi pakan (% dari bobot badan)
50 – 100	3.0
100 – 150	3.5
150 – 200	4.0
200 – 250	3.5
250 – 300	3.0
300 – 350	2.8
350 – 400	2.6
400 – 450	2.4
450 – 500	2.2

**Sumber:** Siregar, 2008

**2.5 Ransum**

Ransum merupakan satu atau beberapa jenis bahan pakan yang diberikan unuk seekor ternak selama sehari semalam. Ransum harus dapat memenuhi zat-zat makanan yang dibutuhkan seekor ternak untuk berbagai fungsi tubuhnya, seperti pokok hidup, produksi, maupun reproduksi (Siregar, 1996).

Pakan merupakan sebuah kebutuhan mutlak yang harus selalu diperhatikan dalam kelangsungan hidup ternak. Pemeliharaan ternak potong dalam skala usaha pokok atau industri memerlukan penyusunan ransum sendiri yang disesuaikan dengan kebutuhan ternak (Santosa, 2006). Oleh karena itu, agar pemberian pakan dapat efisien dan baik dalam pemenuhan biologis, peternak harus memahami tentang kebutuhan zat-zat pakan, bahan pakan ternak, penyusunan ransum, dan daftar susunan zat pakan dari berbagai bahan pakan (Sudarmono, 2008).

Penyusunan ransum diawali dengan mengetahui bobot badan sapi. Bobot badan sapi dapat diketahui secara pasti melalui timbangan. Namun karena timbangan ternak tidak selalu tersedia, maka pengukuran bobot badan sapi dapat dilakukan dengan cara penaksiran atau pengukuran data vital statistik berupa panjang dan lingkaran dada, kemudian dihitung dengan rumus *Schoorl* dan *Lambourne*. Rumus *Schoorl* dapat dituliskan sebagai berikut (Ngadiyono, 2012):

$$\text{Bobot badan (kg)} = \frac{(\text{Lingkaran dada (cm)} + 22)^2}{100} \dots\dots\dots(2-1)$$

Sedangkan *Lambourne* dapat dituliskan sebagai berikut:

$$\text{Bobot badan (kg)} = \frac{\text{panjang badan (cm)} \times \text{lingkaran dada (cm)}^2}{10.840} \dots\dots\dots(2-2)$$

Keterangan (Abidin, 2002):

- Lingkar dada diukur melingkar pada posisi di belakang tonjolan pundak sapi di bagian atas dan bagian belakang kaki depan
- Panjang badan diukur memanjang dari bahu sampai tonjolan tulang duduk

Setelah bobot badan sapi diketahui, kemudian dilanjutkan dengan melakukan perhitungan terhadap kebutuhan pakan sapi. Pada usaha sapi penggemukan, kebutuhan pakan ternak sapi untuk bahan kering (BK) adalah 2,5 – 3 % bobot badan (Ngadiyono, 2012).

Misalkan untuk penyusunan ransum sapi potong jantan yang bobot badannya 250 kg dan mampu pertambahan bobot 1,0 kg/hari. Bahan makan yang tersedia hijauan berupa dedak padi kasar, onggok dan rumput lapangan.

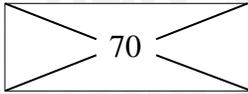
Perhitungan:

Hijauan yang diberikan pada sapi potong tersebut adalah hijauan yang berkualitas sedang. Maka komposisi ransum dalam bentuk bahan kering adalah 60% rumput lapangan dan 40% merupakan campuran dedak padi kasar dan onggok. Kemampuan sapi jantan untuk mengkonsumsi ransum adalah  $250 \times 3,0/100 = 7,5$  kg.

Komposisi ransum diatur sebagai berikut:

- Komposisi rumput lapangan dalam ransum =  $60/100 \times 7,5$  kg = 4,5 kg bahan kering atau  $100/21,8 \times 4,5$  kg = 20,6 kg dalam bentuk segar.
- Komposisi campuran dedak padi kasar dan onggok dalam ransum =  $40/100 \times 7,5$  kg = 3,0 kg bahan kering.
- Energi / TDN dalam rumput lapangan =  $4,5 \times 49/100 \times 1$  kg = 2,2 kg.

Kebutuhan energi/TDN sapi potong tersebut adalah 4,3 kg. Energi yang harus didapatkan dari campuran dedak padi kasar dan onggok adalah  $4,3$  kg -  $2,2$  kg =  $2,1$  kg. Kandungan energi/TDN dedak padi kasar dan onggok =  $2,1/3,0 \times 100\% = 70\%$  dari bahan kering. Selanjutnya digunakan metode bujur sangkar dalam membuat formulasi ransum.

Dedak padi	50		15
Onggok	85		20+
			35

Penjelasan:

- Dedak padi kasar =  $15/35 \times 100\% = 42,9\%$  bahan kering
- Onggok =  $20/35 \times 100\% = 57,1\%$  bahan kering
- Komposisi dedak padi kasar dalam ransum =  $42,9 \times 3,0 \text{ kg} = 1,29 \text{ kg}$  dalam bahan kering atau  $100/87,5 \times 1,29 \text{ kg} = 1,5 \text{ kg}$
- Komposisi onggok dalam ransum =  $57,1 \times 3,0 \text{ kg} = 1,7 \text{ kg}$  dalam bahan kering atau  $100/88,7 \times 1,7 \text{ kg} = 1,9 \text{ kg}$  dalam berat kering

Jadi susunan ransum terdiri dari:

- Dedak padi kasar = 1,5 kg
- Onggok = 1,9 kg
- Rumput lapangan = 20,6 kg

## 2.6 Algoritma Genetika

Algoritma genetika (*Genetic Algorithms*, GAs) merupakan tipe *Evolution Algorithm* (EA) yang paling populer. Algoritma genetika berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat. Karena kemampuannya untuk menyelesaikan berbagai masalah kompleks, algoritma ini banyak digunakan dalam bidang fisika, biologi, ekonomi, sosiologi dan lain-lain yang sering menghadapi masalah optimasi yang model matematikanya kompleks atau bahkan sulit dibangun (Mahmudy, 2013).

Dalam penyelesaian suatu masalah, algoritma genetika memetakan (*encoding*) suatu masalah menjadi string kromosom. String kromosom ini tersusun atas sejumlah *gen* yang menggambarkan variabel-variabel keputusan yang digunakan dalam solusi. Representasi string kromosom beserta fungsi *fitness* untuk menilai seberapa bagus sebuah kromosom untuk menjadi solusi yang layak sehingga dapat dimasukkan ke algoritma genetika (Mahmudy, 2013).

Apabila  $P(t)$  dan  $C(t)$  merupakan populasi (*parents*) dan *offspring* pada generasi ke- $t$  maka struktur algoritma genetika dapat dideksripsikan pada Gambar 2.1 (Gen & Cheng dalam Mahmudy, 2013).

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

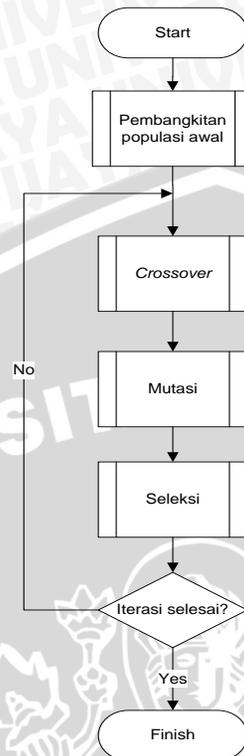
**Gambar 2.1** Pseudocode Algoritma Genetika

**Sumber:** Mahmudy, 2013

Proses dalam algoritma genetika dimulai dengan tahap inisialisasi, yaitu menciptakan individu – individu secara acak yang memiliki susunan gen (kromosom) tertentu yang mewakili solusi dari permasalahan. Tahap selanjutnya adalah reproduksi yang menghasilkan *offspring* dari individu yang ada dipopulasi. Setelah proses reproduksi dilakukan, lahir individu baru yang menyebabkan jumlah individu bertambah. Setiap kromosom mempunyai nilai *fitness*, dimana semakin besar nilai *fitnes* maka semakin baik kromosom tersebut untuk dijadikan calon solusi. Tahap terakhir adalah proses seleksi yaitu memilih individu dari himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya (Mahmudy, 2013).

Setelah melewati sekian iterasi, akan didapatkan individu terbaik. Individu terbaik ini akan mempunyai susunan kromosom yang bisa dikonversi menjadi solusi yang paling baik atau mendekati optimum. Dapat disimpulkan bahwa algoritma genetika menghasilkan suatu solusi optimum dengan melakukan pencarian di antara sejumlah alternatif titik optimum berdasarkan fungsi *probabilistic* (Mahmudy, 2013).

*Flowchart* cara kerja algoritma genetika dapat dilihat pada Gambar 2.2.



**Gambar 2.2** Flowchart Algoritma Genetika

**Sumber:** Aribowo, 2008

### 2.7 Parameter algoritma genetika

Penentuan algoritma genetika merupakan pekerjaan yang tidak mudah. Beberapa parameter algoritma genetika adalah ukuran populasi (*popSize*), banyaknya generasi, *crossover rate* (*cr*), dan *mutation rate* (*mr*). Jika nilai parameter algoritma genetika semakin besar, maka hal ini akan meningkatkan kemampuan eksplorasi algoritma genetika dalam pencarian solusi terbaik. Namun hal ini membuat waktu komputasi berlangsung lama karena kemungkinan algoritma genetika akan mengeksplorasi area yang tidak mempunyai nilai optimum (Mahmudy, 2013).

Tidak adanya metode pasti dalam penentuan nilai parameter algoritma genetika membuat nilai parameter sangat dipengaruhi oleh permasalahan yang akan diselesaikan. Dalam penelitian optimasi menggunakan algoritma genetika, serangkaian pengujian pendahuluan diperlukan untuk mendapatkan kombinasi nilai parameter yang sesuai (Mahmudy, 2013).

## 2.8 Penerapan Algoritma Genetika

Algoritma yang digunakan dalam skripsi ini adalah algoritma genetika dengan pengkodean real (*real-coded genetic algorithms*). Terdapat beberapa tahapan dalam penerapan algoritma genetika, yaitu melakukan representasi kromosom, inisialisasi, reproduksi yang terdiri dari proses *crossover* dan mutasi, evaluasi, lalu yang terakhir adalah proses seleksi. Berikut ini merupakan penjelasan tahapan dalam algoritma genetika.

### 2.8.1 Representasi Kromosom

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Solusi dari suatu permasalahan harus dipetakan (*encoding*) menjadi string kromosom. String kromosom tersusun atas sejumlah gen yang menggambarkan variabel – variabel keputusan yang digunakan dalam solusi (Mahmudy, 2013). Terdapat berbagai cara untuk menentukan representasi kromosom, yaitu sebagai berikut (Imbar & Jayanti, 2011) :

a. Representasi Biner

Representasi yang paling sederhana dan paling umum dimana setiap gen hanya bernilai 0 dan 1, Contoh 1000111, 1000101, 1000100 dan seterusnya.

b. Representasi Integer

Representasi yang bernilai bilangan bulat. Contoh: 29, 18, 21, 9 dan seterusnya.

c. Representasi *Real Code*

Pada penelitian ini, peneliti menggunakan representasi *real code* karena mewakili ukuran dari masing-masing bahan pakan yang akan dioptimasi. Representasi yang membutuhkan tingkat ketelitian amat tinggi representasi ini bernilai bilangan real. Contoh: 65.65 -88.18,21.89 dan seterusnya.

d. Representasi Permutasi

Representasi yang digunakan untuk masalah *scheduling*, *travel salesman problem*, atau yang tidak termasuk dari ketiga representasi.

### 2.8.2 Inisialisasi

Inisialisasi dilakukan untuk membangkitkan himpunan solusi baru secara acak/random yang terdiri dari sejumlah string kromosom dan ditempatkan pada penampungan yang disebut dengan populasi. Dalam tahap ini, ukuran populasi (*popSize*) harus ditentukan. Nilai ini menyatakan jumlah individu/kromosom yang ditampung dalam populasi. Panjang setiap string kromosom (*stringLen*) dihitung berdasarkan presisi variabel dari solusi yang dicari (Mahmudy, 2013). Misalkan populasi inisialisasi dibangkitkan secara random dengan *popSize* = 10 seperti pada Tabel 2.7.

**Tabel 2.7** Contoh Representasi Kromosom

<i>Parent</i>	Kromosom		$f(x_1, x_2)$
	$x_1$	$x_2$	
P1	1,4898	2,0944	19,8206
P2	8,4917	2,5754	34,7058
P3	1,4054	6,3035	20,6707
P4	5,8114	5,0779	14,5624
P5	-1,8461	1,7097	11,5858
P6	4,0206	4,4355	24,7106
P7	-0,1634	2,974	19,653
P8	5,2742	0,7183	22,1813
P9	9,4374	6,6919	12,4694
P10	-4,5575	0,1679	28,4324

**Sumber:** Mahmudy, 2013

### 2.8.3 Reproduksi

Reproduksi bertujuan untuk menghasilkan keturunan dari individu-individu yang ada di populasi. Himpunan keturunan ini akan ditempatkan dalam penampungan *offspring*. Dua operator genetika yang digunakan dalam proses ini adalah *crossover* dan mutasi.

**2.8.3.1 Crossover**

*Crossover* dilakukan dengan cara memilih dua induk (*parent*) secara acak dari populasi. Metode *crossover* yang digunakan adalah *extended intermediate crossover*, yaitu metode yang menghasilkan *offspring* dari kombinasi nilai dua induk. Dalam tahap ini harus ditentukan tingkat *crossover* (*crossover rate / cr*) yang menyatakan rasio *offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak  $cr \times popSize$ .

Misalkan P1 dan P2 adalah dua kromosom yang telah diseleksi untuk melakukan *crossover*, maka bisa dibangkitkan *offspring* C1 dan C2 dengan rumus *offspring* hasil *crossover* sebagai berikut (Mahmudy, 2013):

$$C1 = P1 + \alpha (P2 - P1)$$

$$C2 = P2 + \alpha (P1 - P2) \dots\dots\dots(2-3)$$

Keterangan:

C1, C2 = *Child 1, Child 2*

P1, P2 = *Parent 1, Parent 2*

$\alpha$  = Dipilih secara acak pada range yang ditentukan. Misal pada interval [-0,25; 1,25]

Misalkan yang terpilih sebagai induk adalah P4 dan P9 pada Tabel 2.7,  $\alpha = [0.1104, 1.2336]$  maka akan dihasilkan dua *offspring* (C1 dan C2) sebagai berikut:

$$C1: \quad x1 = 5,8114 + 0,1104 (9,4374 - 5,8114) = 6,2118$$

$$x2 = 5,0779 + 1,2336 (6,6919 - 5,0779) = 7,0690$$

$$C2: \quad x1 = 9,4374 + 0,1104 (5,8114 - 9,4374) = 9,0370$$

$$x2 = 6,6919 + 1,2336 (5,0779 - 6,6919) = 4,700$$

Jika ditentukan  $cr = 0,4$  maka ada  $0,4 \times 10 = 4$  *offspring* yang dihasilkan dari proses *crossover*. Setiap *crossover* akan menghasilkan dua anak, maka terdapat dua kali operasi *crossover* yang akan menghasilkan dua *offspring* berikutnya, yaitu C3 dan C4.

### 2.8.3.2 Mutasi

Mutasi biasanya digunakan sebagai operator untuk menjaga keragaman populasi. Mutasi dilakukan dengan memilih satu induk secara acak dari populasi. Dalam tahap ini nilai tingkat mutasi (*mutation rate / mr*) harus ditentukan untuk menyatakan rasio *offspring* yang dihasilkan dari proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak  $mr \times popSize$  (Mahmudy, 2013).

Metode mutasi yang digunakan adalah *random mutation* yang dilakukan dengan menambah atau mengurangi nilai gen terpilih dengan bilangan *random* yang kecil. Misalkan domain variabel  $x_j$  adalah  $[min_j, max_j]$  dan *offspring* yang dihasilkan adalah  $C=[x'_1..x'_n]$ , maka nilai gen offspring bisa dibangkitkan dengan rumus gen hasil mutasi sebagai berikut (Mahmudy, 2013):

$$x'_i = x'_i + r (max_i - min_j) \dots \dots \dots (2-4)$$

Keterangan:

- $x'_i$  = Induk terpilih
- $max_i$  = Nilai random terbesar
- $min_j$  = Nilai random terkecil
- $r$  = Range, misalkan  $[-0,1, 0,1]$ .

Misal yang terpilih sebagai induk adalah P2 pada Tabel 2.7, gen yang terpilih nomer 2 ( $x_2$ ) dan  $r = -0,0584$ . Maka akan dihasilkan offspring ( $C_5$ ) sebagai berikut:

$$C_5: \quad x_1 = 8,491 \text{ (tetap)}$$

$$x_2 = 2,5754 - 0,0584 (7,3-0,0) = 2,1491$$

Anggap ditentukan  $mr = 0,2$  maka ada  $0,2 \times 10=2$  *offspring* yang dihasilkan dari proses mutasi. *Offspring* dianggap  $C_6$ . Keseluruhan *offspring* yang dihasilkan dari proses reproduksi (*crossover* dan mutasi) dapat dilihat pada Tabel 2.8.

**Tabel 2.8** Hasil *Offspring*

<i>Offspring</i>	<i>Chromosome</i>		$f(x_1, x_2)$
	x1	x2	
C1	6,2118	7,0690	22,2048
C2	9,0370	4,7008	22,2313
C3	7,1636	0,0000	15,4774
C4	7,5479	7,3000	9,3531
C5	8,4917	2,1494	31,0389
C6	-1,1238	1,7097	12,0177

**Sumber:** Mahmudy, 2013

**2.8.4 Evaluasi**

Evaluasi digunakan untuk menghitung *fitness* pada tiap kromosom. Nilai *fitness* merupakan suatu ukuran kualitas dari tiap kromosom. Semakin besar nilai *fitness* maka semakin baik pula kromosom dai himpunan populasi tersebut yang nantinya akan dijadikan calon solusi. Pada kasus dalam pencarian nilai maksimum, nilai *fitness* bisa dihitung langsung dengan rumus *fitness* pencarian nilai maksimum seperti pada persamaan 2-5 (Mahmudy, 2013) :

$$Fitness = f(x) \dots \dots \dots (2-5)$$

Selain pencarian nilai maksimum, *fitness* juga digunakan dalam pencarian nilai minimum. Pada kasus pencarian nilai minimum, nilai *fitness* bisa dihitung dengan salah satu dari dua rumus *fitness* pencarian nilai minimum seperti pada persamaan 2-6 (Mahmudy, 2013):

$$Fitness = C - f(x)$$

$$Fitness = 1 / (f(x)) \dots \dots \dots (2-6)$$

**2.8.5 Seleksi**

Seleksi merupakan tahapan terakhir yang dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar nilai *fitness* kromosom, maka semakin besar peluang kromosom tersebut terpilih. Hal ini dilakukan agar terbentuk generasi berikutnya yang lebih baik dari generasi sekarang (Mahmudy, 2013). Ada beberapa metode seleksi yang dapat digunakan, yaitu:

### a. Seleksi *Elitism*

Metode seleksi yang digunakan dalam skripsi ini adalah metode seleksi *elitism*. Metode seleksi *elitism* bekerja dengan cara mengumpulkan semua individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan.

Metode ini melakukan seleksi pada individu – individu dalam penampungan berdasarkan nilai *fitness* tertinggi. Individu terbaik dalam penampungan akan lolos untuk masuk dalam generasi berikutnya. Metode seleksi *elitism* menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013). *Pseudo-code* seleksi *elitism* dideskripsikan seperti pada Gambar 2.3.

**PROCEDURE ElitismSelection****Input:**

POP: himpunan individu pada populasi

pop\_size: ukuran populasi

OS: himpunan individu anak (*offspring*) hasil reproduksi menggunakan *crossover* dan mutasi**Output :**

POP: himpunan individu pada populasi setelah proses seleksi selesai

/\* gabungkan individu pada POP dan OS ke dalam TEMP \*/

TEMP &lt;- Merge (POP,OS)

/\* urutkan individu berdasarkan *fitness* secara ascending \*/

OrderAscending (Temp)

/\* copy pop\_size individu terbaik ke POP \*/

POP &lt;- CopyBest (Temp, pop\_size)

END PROCEDURE

**Gambar 2.3** Pseudocode Seleksi *Elitism*

**Sumber:** (Mahmudy, 2013)

Himpunan populasi dan *offspring* dicari nilai *fitness*nya masing-masing. Nilai *fitness*nya himpunan populasi dan *offspring* dapat dilihat pada Tabel 2.9.

**Tabel 2.9** Kumpulan Individu

Individu	x1	x2	fitness
P1	1,4898	2,0944	19,8206
P2	8,4917	2,5754	34,7058
P3	1,4054	6,3035	20,6707
P4	5,8114	5,0779	14,5624
P5	-1,8461	1,7097	11,5858
P6	4,0206	4,4355	24,7106
P7	-0,1634	2,974	19,653
P8	5,2742	0,7183	22,1813
P9	9,4374	6,6919	12,4694
P10	-4,5575	0,1679	28,4324
C1	6,2118	7,0690	22,2048
C2	9,0370	4,7008	22,2313
C3	7,1636	0,0000	15,4774
C4	7,5479	7,3000	9,3531
C5	8,4917	2,1494	31,0389
C6	-1,1238	1,7097	12,0177

**Sumber:** Mahmudy, 2013

Metode seleksi *elitism* memilih nilai fitness yang terbesar berdasarkan jumlah *popSize*, sehingga kumpulan individu yang bertahan hidup pada generasi berikutnya dapat dilihat pada Tabel 2.10.

**Tabel 2.10** Individu Hasil Seleksi Elitism

Asal pada P(t)	P(t+1)	Fitness
P1	P1	28,201
C3	P2	23,747
P3	P3	22,496
P4	P4	22,363
P9	P5	21,309
C6	P6	19,656
C1	P7	18,931
P5	P8	18,552
P8	P9	18,077
C5	P10	17,096

**Sumber:** Mahmudy, 2013

b. *Roulette Wheel*

Metode ini menghitung nilai probabilitas seleksi (*prob*) pada tiap individu berdasarkan nilai *fitness*-nya. Nilai *prob* akan menghasilkan probabilitas kumulatif (*probCum*) yang digunakan untuk melakukan proses seleksi. Langkah – langkah membentuk *roulette wheel* berdasarkan probabilitas kumulatif (Mahmudy, 2013) :

- Menghitung total *fitness* keseluruhan dari himpunan populasi *parent* ditambah *offspring*. Misal *fitness* ( $P_i$ ) merupakan nilai *fitness* individu ke –  $i$ . Rumus:

$$\text{total fitness} = \sum_{i=1}^{\text{popSize}} \text{fitness}(P_i) \dots \dots \dots (2-7)$$

- Menghitung nilai probabilitas seleksi (*prob*) tiap individu. Rumus:

$$\text{prob}_k = \frac{\text{fitness}(P_i)}{\text{total fitness}}, \quad i=1, 2, 3, \dots, \text{popSize} \dots \dots \dots (2-8)$$

- Menghitung nilai probabilitas kumulatif tiap individu. Rumus:

$$\text{probCum}_i = \sum_{j=1}^i \text{prob}_j, \quad i=1, 2, 3, \dots, \text{popSize} \dots \dots \dots (2-9)$$

Selanjutnya ketika sudah membentuk *roulette wheel* maka individu akan dipilih secara acak berdasarkan nilai probabilitas kumulatif tersebut.

c. *Binary Tournament Selection*

*Binary Tournament Selection* merupakan metode seleksi dengan melakukan perbandingan individu yang memiliki nilai *fitness* terbaik dari individu – individu yang terpilih secara acak. Individu yang terpilih tersebut maka menjadi individu pada generasi selanjutnya.

Misalkan individu – individu yang terpilih secara acak adalah P1 dan P2. P1 mempunyai nilai *fitness* 12,6342 sedangkan P2 memiliki nilai *fitness* 13,5345. Metode ini akan membandingkan nilai *fitness* antara P1 dan P2. P2 memiliki nilai *fitness* lebih baik daripada P1 sehingga P2 terpilih menjadi individu pada generasi selanjutnya.

#### d. Replacement Selection

Metode ini merupakan metode seleksi dimana *offspring* menggantikan *parent* jika nilai *fitness offspring* lebih besar dari nilai *fitness parent*. Metode ini memiliki aturan berdasarkan cara reproduksinya (Mahmudy, 2013):

- Pada proses mutasi, *offspring* akan menggantikan induknya jika mempunyai nilai *fitness offspring* lebih baik dari *fitness* induknya.
- Pada proses *crossover*, *offspring* dihasilkan dari dua induk. *Offspring* akan menggantikan induk yang terlemah dengan nilai *fitness offspring* lebih baik dari nilai *fitness* induk yang terlemah. Induk yang terlemah merupakan induk yang mempunyai nilai *fitness* terburuk dari dua induk yang menghasilkan *offspring* tersebut.



## BAB III

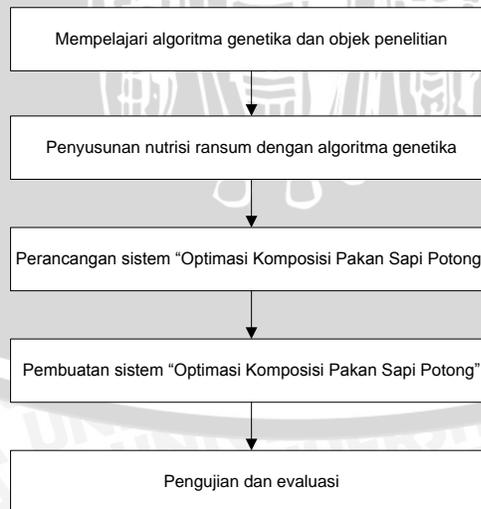
### METODOLOGI PENELITIAN

Pada bab ini akan dibahas beberapa hal, yaitu tahapan penelitian, kebutuhan sistem, formulasi permasalahan, dan siklus penyelesaian masalah menggunakan algoritma genetika.

#### 3.1 Tahapan Penelitian

Pada tahap ini akan dibahas mengenai langkah-langkah dan rancangan yang digunakan dalam pembuatan aplikasi “Optimasi Komposisi Pakan Sapi Potong Menggunakan Algoritma Genetika” sebagai berikut:

1. Melakukan studi literatur mengenai algoritma genetika dalam optimasi pakan sapi potong.
2. Menyusun solusi permasalahan dengan algoritma genetika.
3. Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
4. Implementasi sistem berdasarkan analisis dan perancangan yang dilakukan.
5. Melakukan uji coba dan evaluasi.



**Gambar 3.1** Diagram Blok Tahap Penelitian

### 3.2 Kebutuhan Sistem

Dalam pembuatan aplikasi optimasi distribusi barang dengan algoritma genetika memerlukan beberapa kebutuhan baik berupa kebutuhan perangkat lunak maupun perangkat keras. Berikut ini adalah kebutuhan perangkat lunak dan perangkat keras serta data yang dibutuhkan dalam penelitian :

1. Kebutuhan Perangkat Keras, meliputi :
  - Processor Intel® Core i3-3217U CPU @ 1,80 GHz
  - RAM 4,00 GB
  - Harddisk 500 GB
  - Monitor 14”
2. Kebutuhan Perangkat Lunak, meliputi :
  - Operating System Windows 8.1 64 bit
  - Microsoft Visual Studio Professional 2012 Version 11.0.50727.1
  - Bahasa Pemrograman C#
  - Database MySQL Version 5.5.27

### 3.3 Formulasi Permasalahan

Sub bab ini akan menjelaskan tentang permasalahan yang akan diselesaikan dalam skripsi ini. Permasalahan dalam penelitian ini adalah bagaimana menyusun komposisi pakan sapi potong untuk memenuhi kebutuhan nutrisi dengan biaya pakan yang minimal menggunakan algoritma genetika. Dalam proses menyusun komposisi pakan tersebut, dibutuhkan beberapa data yang meliputi:

- a. Daftar kebutuhan zat gizi sapi jantan untuk pertumbuhan dan penggemukan dari Siregar, Sori Basya (2008) *Penggemukan Sapi*, Jakarta yang terdapat pada Tabel 2.4.
- b. Daftar kebutuhan zat gizi sapi betina untuk pertumbuhan dan penggemukan dari Siregar, Sori Basya (2008) *Penggemukan Sapi*, Jakarta yang terdapat pada Tabel 2.5.

- c. Daftar hasil analisis kimia beberapa bahan pakan konsentrat dari Siregar, Sori Basya (2008) *Penggemukan Sapi*, Jakarta yang terdapat pada Tabel 2.3.
- d. Daftar harga bahan makanan didapatkan berdasarkan survei di Kabupaten Malang pada tahun 2015 yang terdapat pada Lampiran.

Misalkan terdapat seekor sapi potong jantan dengan bobot 250 kg dan pertambahan bobot 1,0 kg/hari. Bahan makan yang tersedia hijauan berupa rumput lapangan, dedak padi kasar, dan onggok. Sesuai pada Tabel 2.4, kebutuhan zat gizi untuk sapi potong jantan yang bobot badannya 250 kg dan pertambahan bobot 1,0 kg/hari adalah sebagai berikut:

- Protein Kasar = 0,76 kg
- Energi/TDN = 4,3 kg

Langkah berikutnya adalah penyusunan ransum dari kombinasi bahan pakan yang tersedia agar dapat menghasilkan bahan pakan yang memenuhi nutrisi minimum dengan biaya optimal menggunakan algoritma genetika. Contoh perhitungan ransum sebagai solusi yang dihasilkan sistem adalah sebagai berikut:

- a. Rumput lapangan

$$\begin{aligned}\text{Bobot pakan} &= 1.5958 \text{ kg} \\ \text{Harga} &= 1.5958 \text{ kg} \times 5000 = \text{Rp. } 7979.02\end{aligned}$$

- b. Dedak padi kasar

$$\begin{aligned}\text{Bobot pakan} &= 1.07506 \text{ kg} \\ \text{Harga} &= 1.07506 \text{ kg} \times 5000 = \text{Rp. } 5375.31\end{aligned}$$

- c. Onggok

$$\begin{aligned}\text{Bobot pakan} &= 4.8291 \text{ kg} \\ \text{Harga} &= 4.8291 \text{ kg} \times 4000 = \text{Rp. } 19316.53\end{aligned}$$

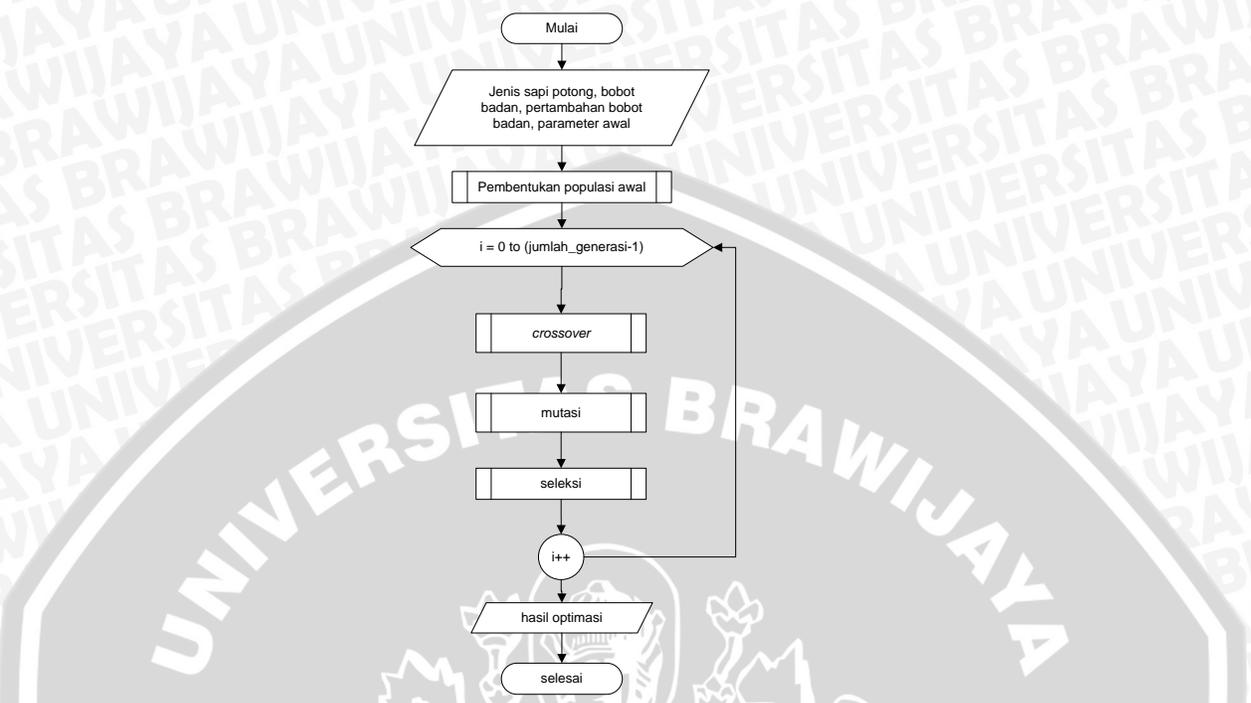
$$\begin{aligned}\text{Total Harga} &= \text{harga rumput lapangan} + \text{harga dedak padi kasar} + \text{harga onggok} \\ &= \text{Rp. } 7979.02 + \text{Rp. } 5375.31 + \text{Rp. } 19316.53 \\ &= \text{Rp. } 32670.87\end{aligned}$$

### 3.4 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

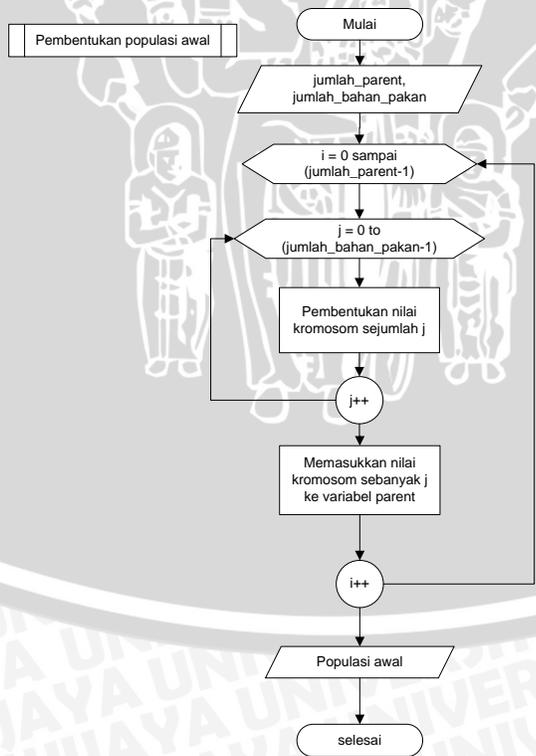
Langkah awal proses ini adalah penentuan jenis sapi potong, bobot badan sapi potong, penentuan penambahan bobot badan sapi potong, bahan pakan yang tersedia, dan parameter awal. Langkah selanjutnya adalah melakukan inisialisasi populasi awal. Representasi kromosom didapat dari pembangkitan nilai secara acak antara 0-10 karena sistem mempunyai nilai keluaran berbentuk jumlah bobot pakan diantara *range* 0-10. Dalam tahap ini, ukuran populasi (*popSize*) harus ditentukan. Nilai ini menyatakan jumlah individu / kromosom yang ditampung dalam populasi. Inisialisasi parameter awal ditentukan sebagai berikut:

- a. Jumlah populasi (*popSize*) = 10
- b. *Crossover rate* (*cr*) = 0.4
- c. *Mutation rate* (*mr*) = 0.1
- d. Banyaknya generasi = 1

Populasi awal dibangkitkan dari nilai parameter yang dimasukkan. Proses reproduksi dilakukan dengan *crossover* dan mutasi. *Offspring* dihasilkan dari jumlah *popSize* x *cr* (*crossover rate*) dan *popSize* x *mr* (*mutation rate*). Selanjutnya individu dari populasi awal dan *offspring* hasil *crossover* dan mutasi digabungkan untuk proses seleksi. Proses seleksi dilakukan dengan menghitung nilai *fitness* masing-masing individu. Individu yang mampu bertahan adalah individu yang memiliki nilai *fitness* terbaik setelah *n* generasi. Nilai *fitness* didapat dari harga bahan pakan dan penalti yang didapat, yaitu jumlah nutrisi yang kurang dari bahan pakan tersebut. *Flowchart* perhitungan formulasi ransum akan dijelaskan pada Gambar 3.2 dan *flowchart* sub proses pembentukan populasi awal akan dijelaskan pada Gambar 3.3.



Gambar 3.2 Flowchart Perhitungan Ransum dengan Algoritma Genetika



Gambar 3.3 Flowchart Subproses Pembentukan Populasi Awal

### 3.4.1 Representasi Kromosom dan Perhitungan Fitness

Jumlah populasi awal ditentukan dari jumlah populasi yang telah diinputkan. Populasi awal dibangkitkan dengan cara bahwa satu bahan pakan dianggap sebagai satu gen penyusun individu. Jika terdapat empat bahan pakan, maka panjang kromosom adalah empat.

Pembentukan nilai kromosom inisial dilakukan dengan membangkitkan nilai *random* pada interval yang telah ditentukan yaitu diantara 0-10. Representasi kromosom dari tiga bahan pakan yang tersedia rumput lapangan, dedak padi kasar, dan onggok dapat dilihat pada Tabel 3.1

**Tabel 3.1** Populasi Awal

<i>Parent</i>	<b>Kromosom</b>		
	Rumput lapangan	Dedak padi kasar	Onggok
1	7.102252	2.977962	9.769518
2	9.303361	2.027727	4.739411
3	4.591873	0.2822434	4.034647
4	5.593279	5.895082	9.416166
5	1.200974	4.662406	5.803459
6	1.979451	1.333518	5.990102
7	5.160055	5.289682	9.323572
8	1.06777	5.702681	1.913983
9	1.462714	6.627262	5.572344
10	4.740237	0.2785558	1.157407

Berdasarkan data pada Tabel 2.4, seekor sapi potong jantan yang bobot badannya 250 kg dan pertambahan bobot 1,0 kg/hari membutuhkan protein kasar sebesar 0,76 kg dan energi/TDN sebesar 4,3 kg. Berdasarkan Tabel 2.6, sapi dengan bobot 250 kg membutuhkan bahan pakan sebesar 3 x bobot, maka  $3 \times 250 \text{ kg} = 7.5 \text{ kg}$ .

Berikut adalah contoh perhitungan nilai *fitness* untuk individu pertama yang diambil dari Tabel 3.1 untuk *Parent 1* adalah sebagai berikut:

Panjang gen masing-masing bahan pakan:

- Rumput lapangan = 7.102252
- Dedak padi kasar = 2.977962
- Onggok = 9.769518

Maka total kromosom = 7.102252 + 2.977962 + 9.769518 = 19.849732

Menghitung nutrisi bahan pakan dilakukan dengan menghitung protein kasar (PK) dan energi / TDN untuk masing-masing bahan pakan yang tersedia dengan melihat Tabel 2.3. Perhitungan kandungan nutrisi masing-masing bahan pakan dilakukan sesuai dengan rumus penentuan kandungan nutrisi pada persamaan 3-1.

$$\text{Kandungan nutrisi} = \frac{\text{gen pakan}}{\text{Total kromosom}} \times \text{bobot pakan akhir} \times \text{nutrisi} \dots\dots(3-1)$$

- a. Kandungan nutrisi rumput lapangan
  - PK = (7.102252 / 19.849732) x 7.5 x 6.7 = 0.179794953
  - TDN = (7.102252 / 19.849732) x 7.5 x 56.2 = 1.508130799
- b. Kandungan nutrisi dedak padi kasar
  - PK = (2.977962 / 19.849732) x 7.5 x 13.8 = 0.155276186
  - TDN = (2.977962 / 19.849732) x 7.5 x 5.5 = 0.061885436
- c. Kandungan nutrisi onggok
  - PK = (9.769518 / 19.849732) x 7.5 x 1.8 = 0.066443463
  - TDN = (9.769518 / 19.849732) x 7.5 x 85 = 3.137607966

PK dan TDN untuk masing-masing bahan pakan diuraikan pada Tabel 3.2.

**Tabel 3.2** Nilai PK dan TDN Masing-Masing Bahan

Rumput lapangan		Dedak padi kasar		Onggok	
PK	TDN	PK	TDN	PK	TDN
0.179794953	1.508130799	0.155276186	0.061885436	0.066443463	3.137607966

Langkah selanjutnya adalah menjumlahkan PK dan TDN masing-masing bahan

- PK = 0.179794953 + 0.155276186 + 0.066443463 = 0.401514601
- TDN = 1.508130799 + 0.061885436 + 3.137607966 = 4.707624201

Setelah menjumlahkan nutrisi, yaitu PK dan TDN, selanjutnya adalah melakukan perhitungan *penalty* dengan cara menghitung nilai mutlak dari selisih antara ketersediaan nutrisi dan kebutuhan nutrisi sapi yang dapat dilihat pada tabel 2.4 yaitu PK sebesar 0.76 dan TDN sebesar 4.3.

- *Penalty* PK = | Kebutuhan – Ketersediaan |  
= | 0.76 - 0.401514601 | = 0.358485399
- *Penalty* TDN = | Kebutuhan – Ketersediaan |  
= | 4.3 - 4.707624201 | = 4.07624201
- Total *Penalty* = *Penalty* PK + *Penalty* TDN  
= 0.358485399 + 4.07624201 = 0.76610960

Kebutuhan gizi dan ketersediaan gizi bahan pakan dapat dilihat pada Tabel 3.3.

**Tabel 3.3** Kebutuhan dan Ketersedian PK dan TDN

Kebutuhan		Ketersediaan		Total <i>Penalty</i>
PK	TDN	PK	TDN	
0,76	4.3	0.401514601	4.707624201	0.76610960

Langkah berikutnya adalah menghitung total harga setiap bahan pakan dengan cara mengalikan bobot bahan dengan harga per-kilogram. Harga yang didapatkan diambil dari Lampiran. Penentuan harga pakan dilakukan sesuai perhitungan pada rumus penentuan harga pada persamaan 3-2.

$$\text{Harga} = \frac{\text{gen pakan}}{\text{Total kromosom}} \times \text{bobot pakan akhir} \times \text{harga bahan pakan} \dots (3-2)$$

a. Harga rumput lapangan

$$\text{Harga} = (7.102252 / 19.849732) \times 7.5 \times 5000 = 13417.5338$$

b. Harga dedak padi kasar

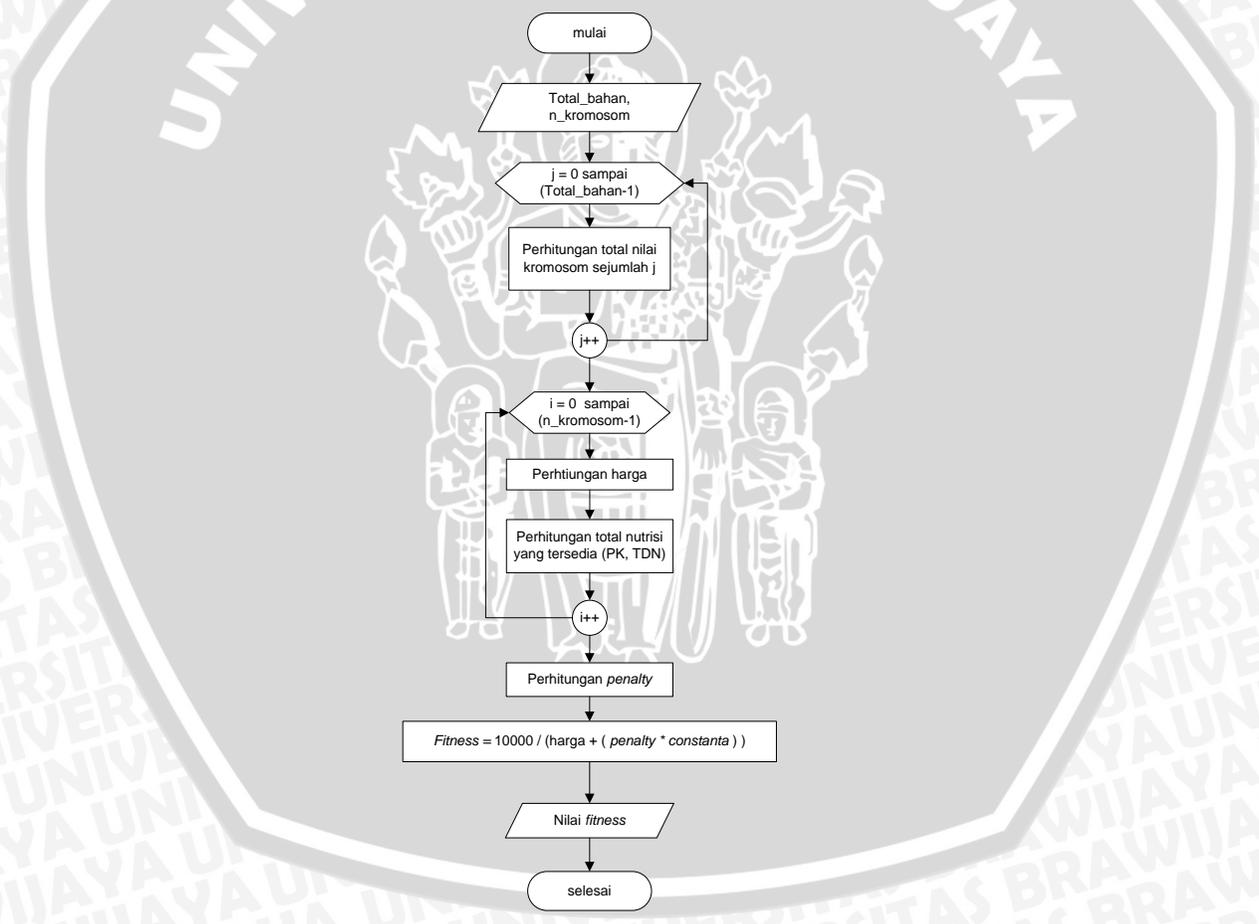
$$\text{Harga} = (2.977962 / 19.849732) \times 7.5 \times 5000 = 5625.948753$$

c. Harga Ongkok

$$\text{Harga} = (9.769518 / 19.849732) \times 7.5 \times 4000 = 14765.21396$$

$$\begin{aligned} \text{Total Harga} &= \text{harga rumput lapangan} + \text{harga dedak padi kasar} + \text{harga ongkok} \\ &= 13417.5338 + 5625.948753 + 14765.21396 = 33808.69651 \end{aligned}$$

Setelah menghitung harga, maka berikutnya adalah menghitung nilai *fitness*. Flowchart proses perhitungan nilai *fitness* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Perhitungan *Fitness*

Susunan kromosom yang membentuk suatu individu terdiri dari macam-macam bahan pakan. Masing-masing bahan pakan tersebut memiliki bobot yang berbeda. Proses perhitungan nilai *fitness* ditentukan dari total harga dari setiap bahan pakan dan penalti dari setiap kromosom. Perhitungan nilai *fitness* menggunakan rumus *fitness* pada persamaan 3-3.

$$Fitness = \frac{10000}{\text{harga} + (\text{penalty} \times C)} \dots\dots\dots(3-3)$$

Keterangan :

- Harga : Total harga dari semua bahan pakan
- *Penalty* : Nilai gizi yang kurang dari yang dibutuhkan
- *C* : Nilai pengali untuk menghilangkan *penalty*, sebesar 5000 karena menyesuaikan dengan rata-rata harga bahan pakan.

Pada kasus komposisi pakan sapi potong ini, nilai konstanta *C* yang digunakan adalah sebesar 5000. Nilai 5000 didapatkan dari rentang harga biaya pakan yang terdapat pada Lampiran. Hal ini dilakukan dengan tujuan agar nilai *penalty* yang sebelumnya sangat kecil dapat memiliki nilai yang sebanding dengan nilai harga. Pada akhirnya, harga dan *penalty* memiliki tingkat kepentingan yang sama dengan tujuan agar individu yang terpilih sebagai solusi merupakan individu yang fokus pada proses meminimalkan nilai *penalty* dan harga biaya pakan.

Sehingga didapatkan nilai *fitness* sebagai berikut:

$$Fitness = \frac{10000}{33808.69651 + (0.76610960 \times 5000)} = 0.26568015$$

Hasil nilai *fitness* dapat dilihat pada Tabel 3.4.

**Tabel 3.4** Nilai *Fitness*

Kromosom			Nilai Fitness
1	2	3	
Rumput lapangan	Dedak padi kasar	Onggok	0.26568015
7.102252	2.977962	9.769518	

### 3.4.2 Reproduksi

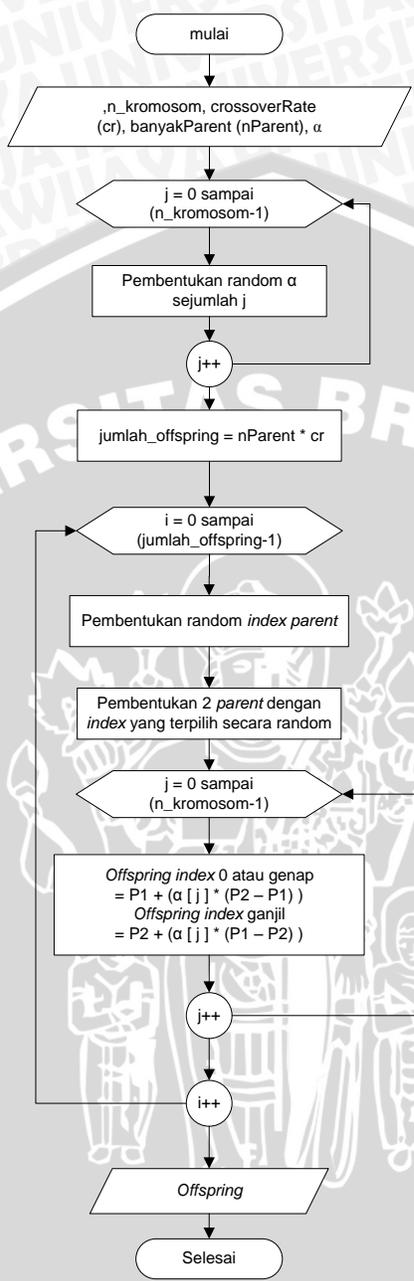
Reproduksi dilakukan untuk mendapatkan keturunan dari individu-individu dalam populasi. Himpunan keturunan nantinya ditempatkan dalam penampungan *offspring*. Pada saat reproduksi terdapat dua operator genetika yang digunakan yaitu *crossover* dan proses mutasi (Mahmudy, 2013).

Metode yang digunakan dalam proses *crossover* adalah metode *extended intermediate crossover* sedangkan metode yang digunakan dalam proses mutasi adalah metode *random mutation*.

#### 3.4.2.1 Perhitungan *Crossover*

Metode *crossover* yang digunakan adalah *extended intermediate* yang menghasilkan *offspring* dari kombinasi nilai dua induk. Dalam tahap ini harus ditentukan tingkat *crossover* (*crossover rate / cr*) yang menyatakan rasio *offspring* yang dihasilkan proses *crossover* terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak  $cr \times popSize$ .

Misalkan P1 dan P2 adalah dua kromosom yang telah diseleksi untuk melakukan *crossover*, maka *Offspring1* dan *Offspring2* bisa dibangkitkan dengan persamaan (2-3) (Mahmudy, 2013). Flowchart dalam melakukan proses *crossover* dapat dilihat pada Gambar 3.5



**Gambar 3.5** Flowchart Proses *Crossover*

Banyaknya *offspring* yang dihasilkan pada proses *crossover* adalah  $cr \times popSize = 0.4 \times 10 = 4$ . Nilai  $\alpha$  dipilih secara acak pada interval  $[-0.25, 1.25]$  yakni 0.364663935, 0.854051416, dan 0.353361207. Misalkan, *parent* acak yang terpilih untuk proses *crossover* pertama adalah *Parent* 10 dan *Parent* 3, sedangkan untuk

proses *crossover* kedua, *parent* acak yang terpilih adalah *Parent 2* dan *Parent 10*.

Hasil perhitungan manual *crossover* dapat dilihat pada Tabel 3.5.

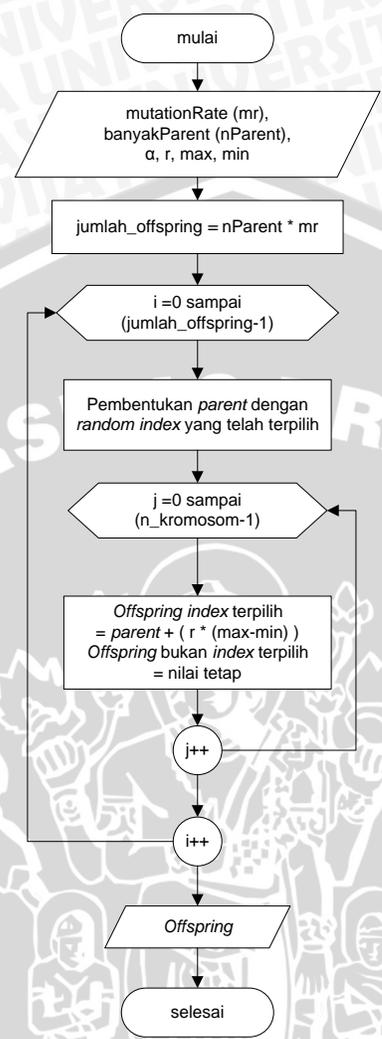
**Tabel 3.5** Perhitungan Manual *Crossover*

Individu	Kromosom		
<i>Parent 10</i>	4.740237	0.2785558	1.157407
<i>Parent 3</i>	4.591873	0.2822434	4.034647
<i>Offspring 1</i>	4.686134	0.2817052	2.174112
<i>Offspring 2</i>	4.645976	0.279094	3.017942
<i>Parent 2</i>	9.303361	2.027727	4.739411
<i>Parent 10</i>	4.740237	0.2785558	1.157407
<i>Offspring 3</i>	7.639362	0.5338444	3.47367
<i>Offspring 4</i>	6.404236	1.772438	2.423148

### 3.3.2.2 Perhitungan Mutasi

Mutasi dilakukan dengan memilih satu induk secara acak dari populasi. *Random mutation* digunakan sebagai metode mutasi yang dilakukan dengan menambah atau mengurangi nilai gen terpilih dengan bilangan *random* yang kecil seperti pada Persamaan (2-4).

Dalam tahap ini nilai tingkat mutasi (*mutation rate / mr*) harus ditentukan untuk menyatakan rasio *offspring* yang dihasilkan dari proses mutasi terhadap ukuran populasi sehingga akan dihasilkan *offspring* sebanyak  $mr \times popSize$  (Mahmudy, 2013). *Flowchart* dalam melakukan proses mutasi dapat dilihat pada Gambar 3.6



**Gambar 3.6** Flowchart Proses Mutasi

Banyaknya *offspring* yang dihasilkan pada proses mutasi adalah  $mr \times popSize = 0,1 \times 10 = 1$ . Nilai  $r$  berada pada interval  $[-0,1, 0,1]$  yakni  $-0.7102252$  dan *parent* yang terpilih adalah *parent* 1 dengan gen nomor 1. Maka akan dihasilkan *offspring* pada Tabel 3.6.

**Tabel 3.6** Perhitungan Manual Mutasi

Individu	Kromosom		
Parent 1	7.102252	2.977962	9.769518
Offspring 5	6.653904	2.977962	9.769518

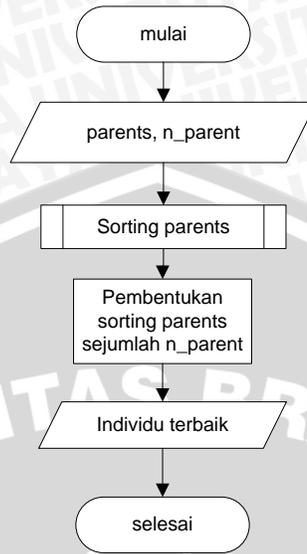
### 3.4.3 Evaluasi dan Seleksi

Setelah proses *crossover* dan mutasi selesai dilakukan, selanjutnya dilakukan proses evaluasi. Proses evaluasi dilakukan dengan menghimpun seluruh individu yaitu *parent* dan *offspring* dan dilakukan perhitungan nilai *fitness* masing-masing individu. Semakin besar nilai *fitness* suatu individu maka semakin besar peluang untuk lolos (Gen & Cheng dalam Mahmudy, 2013). Perhitungan nilai *fitness* menggunakan Persamaan (3-1) sehingga didapatkan Tabel 3.7.

**Tabel 3.7** Hasil Perhitungan *Fitness* Seluruh Individu

Individu	Kromosom			Tersedia		Total Penalti	Harga	Fitness
	Rumput Lapangan	Dedak Padi Kasar	Onggok	PK	TDN			
<i>Parent 1</i>	7.1023	2.9780	9.7695	0.40151	4.70762	0.76611	33808.70	0.265680
<i>Parent 2</i>	9.3034	2.0277	4.7394	0.46131	4.37223	0.37092	35288.15	0.269232
<i>Parent 3</i>	4.5919	0.2822	4.0346	0.35294	5.07276	1.17983	34103.36	0.249984
<i>Parent 4</i>	5.5933	5.8951	9.4162	0.48713	4.11564	0.45723	34121.72	0.274666
<i>Parent 5</i>	1.2010	4.6624	5.8035	0.53250	3.76986	0.75764	33769.26	0.266259
<i>Parent 6</i>	1.9795	1.3335	5.9901	0.34220	5.06073	1.17853	32670.87	0.259312
<i>Parent 7</i>	5.1601	5.2897	9.3236	0.47167	4.21626	0.37207	33963.58	0.279143
<i>Parent 8</i>	1.0678	5.7027	1.9140	0.77117	2.19412	2.11706	35847.06	0.215367
<i>Parent 9</i>	1.4627	6.6273	5.5723	0.61091	3.25148	1.19761	34441.03	0.247347
<i>Parent 10</i>	4.7402	0.2786	1.1574	0.45765	4.44828	0.45063	36094.52	0.260772
<i>Offspring 1</i>	4.6861	0.2817	2.1741	0.41163	4.72255	0.77092	35216.89	0.255941
<i>Offspring 2</i>	4.6460	0.2791	3.0179	0.38158	4.90208	0.98050	34650.38	0.252826
<i>Offspring 3</i>	7.6394	0.5338	3.4737	0.41730	4.68493	0.72763	35263.13	0.257061
<i>Offspring 4</i>	6.4042	1.7724	2.4231	0.50753	4.07295	0.47952	35785.48	0.261896
<i>Offspring 5</i>	6.6539	2.9780	9.7695	0.39918	4.71901	0.77983	33723.39	0.265798

Seleksi dilakukan dengan cara memilih individu dari himpunan populasi dan *offspring* yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar nilai *fitness* kromosom, maka semakin besar peluang kromosom tersebut terpilih. Flowchart proses seleksi dengan metode *elitism* dapat dilihat pada Gambar 3.7.



**Gambar 3.7** Flowchart Proses Seleksi *Elitism*

Metode seleksi yang digunakan adalah *elitism*. Metode ini melakukan seleksi pada individu – individu dalam penampungan berdasarkan nilai *fitness* tertinggi. Individu terbaik dalam penampungan akan lolos untuk masuk dalam generasi berikutnya (Mahmudy, 2013). Hasil seleksi *elitism* dapat dilihat pada Tabel 3.8.

**Tabel 3.8** Hasil Seleksi *Elitism*

Individu	<i>Fitness</i>
Parent 7	0.27914293
Parent 4	0.27466574
Parent 2	0.26923167
Parent 5	0.26625870
Offspring 5	0.26579817
Parent 1	0.26568015
Offspring 4	0.26189613
Parent 10	0.26077195
Parent 6	0.25931240
Offspring 3	0.25706107

Setelah dilakukan proses seleksi, selanjutnya dilakukan pemilihan individu terbaik berdasarkan nilai *fitness* tertinggi. Hasil kromosom terbaik pada generasi awal dapat dilihat pada Tabel 3.9.

**Tabel 3.9** Pemilihan Kromosom Terbaik

Individu	Kromosom			Tersedia		Total Penalti	Harga	Fitness
	1	2	3	PK	TDN			
	Rumput lapangan	Dedak padi kasar	Onggok					
Parent 7	5.16006	5.28968	9.32357	0.47167	4.21626	0.37207	33963.58	0.279143

Berdasarkan Tabel 2.6, sapi dengan bobot 250 kg membutuhkan bahan pakan sebesar 3 x bobot, maka  $3 \times 250 \text{ kg} = 7.5 \text{ kg}$ . Total kromosom yang dihasilkan adalah  $5.16006 + 5.28968 + 9.32357 = 19.773309$ . Maka ransum optimal yang diberikan sistem memiliki berat masing-masing bahan pakan adalah sebagai berikut:

a. Rumput lapangan

$$\text{Bobot pakan} = 5.16006 / 19.773309 \times 7.5 \text{ kg} = 1.957205 \text{ kg}$$

$$\text{Harga} = 1.957205 \text{ kg} \times 5000 = \text{Rp. } 9786.02$$

b. Dedak padi kasar

$$\text{Bobot pakan} = 5.28968 / 19.773309 \times 7.5 \text{ kg} = 2.006372 \text{ kg}$$

$$\text{Harga} = 2.006372 \text{ kg} \times 5000 = \text{Rp. } 10031.86$$

c. Onggok

$$\text{Bobot pakan} = 9.32357 / 19.773309 \times 7.5 \text{ kg} = 3.536423 \text{ kg}$$

$$\text{Harga} = 3.536423 \text{ kg} \times 4000 = \text{Rp. } 14145.69$$

$$\text{Total Harga} = \text{harga rumput lapangan} + \text{harga dedak padi kasar} + \text{harga onggok}$$

$$= \text{Rp. } 9786.02 + \text{Rp. } 10031.86 + \text{Rp. } 14145.69$$

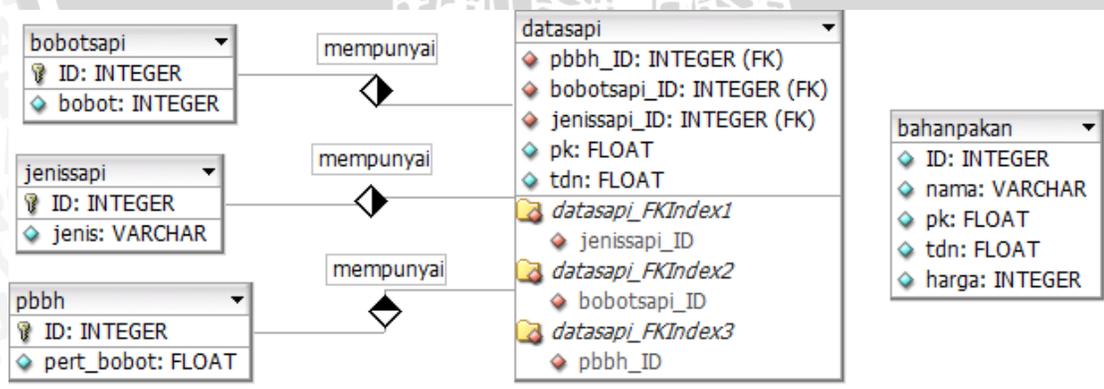
$$= \text{Rp. } 33963.58$$

## BAB IV PERANCANGAN SISTEM

Pada bab ini akan dibahas beberapa hal, yaitu perancangan *database*, perancangan antarmuka aplikasi dan yang terakhir adalah perancangan uji coba dan evaluasi.

### 4.1 Perancangan Database

Pada tahap perancangan *database*, tabel yang digunakan ada lima, yaitu bobotsapi, jenisapi, pbbh, datasapi, dan bahanpakan. Tabel bobotsapi memiliki dua field yaitu ID sebagai *primary key* dan bobot. Tabel jenis sapi memiliki field diantaranya ID sebagai *primary key* dan jenis. Tabel pbbh memiliki dua field yaitu ID sebagai *primary key* dan pert\_bobot. Tabel datasapi memiliki lima field, yaitu pk dan tdn yang berbentuk float, sedangkan tiga sisanya merupakan *foreign key* yang berasal dari tabel bobotsapi, jenisapi, dan pbbh. Tabel bahanpakan memiliki lima field yaitu ID dan harga yang berbentuk integer, nama yang berbentuk varchar, serta pk dan tdn yang berbentuk float. Tabel bahanpakan tidak berelasi karena hanya digunakan untuk menyimpan data. Desain *database* dapat dilihat pada Gambar 4.1.



**Gambar 4.1** Desain Database

## 4.2 Perancangan Antarmuka Aplikasi

Antarmuka aplikasi ini terdiri dari tiga halaman. Halaman pertama adalah halaman utama yang terdiri dari inputan pengguna, detail perhitungan, dan hasil rekomendasi. Halaman kedua adalah halaman untuk memasukkan bahan pakan yang tersedia. Halaman terakhir adalah halaman untuk menentukan parameter algoritma genetika.

### 4.2.1 Rancangan Halaman Utama

Halaman ini merupakan halaman utama yang digunakan pengguna untuk memasukkan data yang dibutuhkan oleh sistem, seperti data sapi, data parameter, dan data bahan pakan. Masukan pengguna akan diolah program yang akan menghasilkan keluaran dan ditampilkan di detail perhitungan, kromosom terbaik, dan rekomendasi. Rancangan halaman utama ditampilkan pada Gambar 4.2.

The wireframe shows a web page layout for a genetic algorithm-based feed rationing application. The page is titled "1 RANSUM SAPI POTONG ALGORITMA GENETIKA". It is divided into two main sections: "INPUT DATA SAPI POTONG" and "INPUT DATA BAHAN PAKAN".

- INPUT DATA SAPI POTONG:** Contains a "5 Parameter Setting" button, and three input fields labeled "2" (Jenis Sapi), "3" (Bobot Badan), and "4" (Pertambahan Bobot Badan).
- INPUT DATA BAHAN PAKAN:** Contains "6" (Tambah) and "7" (Hapus) buttons, and a large area labeled "8".
- 9 HITUNG:** A prominent button to initiate the calculation.
- Output Areas:** "10 KROMOSOM TERBAIK" (Best Chromosomes) and "11 DETAIL PERHITUNGAN" (Calculation Details).
- 12 REKOMENDASI:** A section for displaying the final feed recommendations.

Gambar 4.2 Rancangan Halaman Utama

Keterangan:

1. Judul aplikasi
2. *Textbox* untuk memasukkan jenis sapi
3. *Textbox* untuk memasukkan bobot sapi
4. *Textbox* untuk memasukkan penambahan bobot sapi
5. Tombol start untuk membuka form parameter
6. Tombol tambah untuk memilih bahan pakan yang tersedia
7. Tombol hapus untuk menghapus bahan pakan yang telah dipilih
8. *List* bahan pakan yang telah dipilih
9. Tombol hitung untuk memulai perhitungan
10. *List* kromosom terbaik yang dihasilkan aplikasi
11. Detail perhitungan algoritma genetika
12. Rekomendasi pakan kepada pengguna

#### 4.2.2 Rancangan Halaman Bahan Pakan

Halaman ini merupakan halaman yang digunakan pengguna untuk menginputkan bahan pakan yang tersedia. Rancangan halaman bahan pakan ditampilkan pada Gambar 4.3.

Daftar Bahan Pakan				
ID	Bahan Pakan	Protein Kasar	Energi / TDN	Harga
		1		

2   3

Gambar 4.3 Rancangan Halaman Bahan Pakan

Keterangan:

1. *List* bahan pakan yang tersedia pada *database*
2. Tombol tambah untuk menambahkan bahan pakan ke dalam halaman utama
3. Tombol batal untuk membatalkan bahan pakan yang telah dipilih sebelumnya

#### 4.2.3 Rancangan Halaman Parameter

Halaman ini merupakan halaman yang digunakan pengguna untuk menentukan parameter yang digunakan untuk perhitungan algoritma genetika. Rancangan halaman parameter ditampilkan pada Gambar 4.4.

The image shows a dialog box titled "Parameter" with a white background and a black border. It contains five numbered items:

- 1. "Jumlah Parent" with a text input field.
- 2. "Crossover rate (cr)" with a combobox.
- 3. "Mutation rate (mr)" with a combobox.
- 4. "Jumlah Generasi" with a text input field.
- 5. An "OK" button.

**Gambar 4.4** Rancangan Halaman Parameter

Keterangan:

1. *Textbox* untuk memasukkan jumlah parent
2. *Combobox* untuk memasukkan *crossover rate*
3. *Combobox* untuk memasukkan *mutation rate*
4. *Textbox* untuk memasukkan banyaknya generasi
5. Tombol untuk memasukkan parameter algoritma genetika ke dalam sistem

### 4.3 Perancangan Uji Coba dan Evaluasi

Tidak adanya metode tertentu untuk menentukan parameter algoritma genetika membuat perlu dilakukannya evaluasi program dengan uji coba agar mendapatkan parameter yang optimal. Uji coba tersebut antara lain:

1. Uji coba untuk menentukan ukuran populasi yang optimal.
2. Uji coba untuk menentukan banyaknya generasi yang optimal.
3. Uji coba untuk mencari kombinasi *crossover rate* dan *mutation rate* terbaik untuk menghasilkan rekomendasi yang optimal.

#### 4.3.1 Uji Coba Ukuran Populasi

Uji coba ukuran populasi yaitu uji coba yang digunakan untuk mengetahui ukuran populasi yang tepat agar dapat menghasilkan komposisi pakan sapi potong yang optimal. Banyaknya populasi yang digunakan adalah kelipatan 50. Rancangan uji coba populasi dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Pemilihan Kromosom Terbaik

Banyak populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan generasi ke-										
	1	2	3	4	5	6	7	8	9	10	
50											
100											
150											
200											
250											
300											
350											
400											
450											

### 4.3.2 Uji Coba Banyaknya Generasi

Uji coba generasi adalah uji coba yang dilakukan untuk mengetahui banyaknya generasi yang dibutuhkan agar dapat mendapatkan hasil komposisi bahan pakan sapi potong yang optimal. Banyaknya generasi yang di uji coba adalah dimulai dari 50 dan mempunyai kelipatan 25. Rancangan uji coba banyaknya generasi dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Rancangan Uji Coba Banyaknya Generasi

Banyaknya generasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	percobaan generasi ke-										
	1	2	3	4	5	6	7	8	9	10	
50											
75											
100											
125											
150											
175											
200											
225											
250											

### 4.3.3 Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Uji coba kombinasi *crossover rate* (cr) dan *mutation rate* (mr) adalah uji coba yang dilakukan untuk mengetahui kombinasi cr dan mr terbaik untuk mendapatkan hasil komposisi bahan pakan sapi potong yang optimal. Uji coba ini menggunakan nilai cr dan mr yang berbeda antara *range* 0 - 0.6. Rancangan uji coba kombinasi *crossover rate* dan *mutation rate* dapat dilihat pada Tabel 4.3.

**Tabel 4.3** Rancangan Uji Coba Kombinasi Cr dan Mr

Kombinasi		Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
		percobaan generasi ke-										
Cr	Mr	1	2	3	4	5	6	7	8	9	10	
0.6	0											
0.5	0.1											
0.4	0.2											
0.3	0.3											
0.2	0.4											
0.1	0.5											
0	0.6											



## BAB V

### IMPLEMENTASI SISTEM

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang telah dibuat. Pembahasan pada bab ini terdiri dari implementasi program dan implementasi antarmuka aplikasi “Optimasi Komposisi Pakan Sapi Potong Menggunakan Algoritma Genetika”.

#### 5.1 Implementasi Program

Berdasarkan perancangan perangkat lunak yang telah dibahas pada bab 3, maka akan dibahas mengenai implementasi program sesuai dengan perancangan yang telah dibuat. Sistem diimplementasikan dengan menggunakan bahasa pemrograman C# berbasis desktop.

##### 5.1.1 Implementasi Proses untuk Inisialisasi Populasi Awal

Proses ini diawali dengan inputan pengguna berupa jenis sapi potong, bobot badan sapi potong, pertumbuhan bobot badan, bahan pakan yang tersedia, cr, mr, *popSize*, dan jumlah iterasi. Setelah inputan selesai dimasukkan, proses inisialisasi populasi awal dimulai dengan membangkitkan kromsوم tiap individu secara random antara 0 - 10. Proses inisialisasi dapat dilihat pada *Source Code* 5.1.

```
1 private void generateParent()  
2     {  
3         if (parents != null)  
4             {  
5                 parents.Clear();  
6                 parents = null;  
7                 System.GC.Collect();  
8                 tampilText = "";  
9             }  
10  
11     parents = new List<Parent>();
```

```

12 tampilText += "<b>Nutrisi yang dibutuhkan
13 sapi</b><br>";
14 tampilText += "Protein kasar : <b>" + sapi.pk + "</b>
15 <br>";
16 tampilText += "Energi / TDN : <b>" + sapi.tdn + "</b>
17 <br>";
18 for (int i = 0; i < nParent; i++)
19 {
20     Parent par = new Parent(sapi);
21     for (int j = 0; j < parent.getTotalBahan(); j++)
22     {
23         par.addBahanPakan((BahanPakan)parent.getBahanPak
24         an(j).Clone());
25         par.getBahanPakan(j).setPanjangStringKromosom(Ne
26         xtFloat(random));
27     }
28     par.kalkulasi(bobot, pakanharian);
29     par.nama = "P" + (i + 1);
30     parents.Add(par);
31 }
32 }

```

#### **Source Code 5.1** Inisialisasi Populasi Awal

Penjelasan dari *source code* pada Source Code 5.1 adalah sebagai berikut:

1. Baris 3 - 9 merupakan proses untuk mengosongkan nilai parents
2. Baris 11 merupakan proses inisialisasi variabel
3. Baris 12 – 17 merupakan proses untuk menampilkan jumlah nutrisi yang dibutuhkan sapi potong
4. Baris 18 – 27 merupakan proses untuk menghasilkan nilai random awal dengan panjang sesuai jumlah bahan pakan.
5. Baris 28 merupakan proses untuk menghitung nutrisi, harga, *penalty*, dan *fitness*
6. Baris 29 merupakan proses penamaan *parent*
7. Baris 30 merupakan proses penambahan variabel par yang berisi nilai random awal ke dalam variabel parents

### 5.1.2 Implementasi Proses untuk Perhitungan *Fitness*

Proses perhitungan *fitness* dilakukan dengan cara memasukkan persamaan *fitness* yang telah ditentukan sebelumnya kepada individu yang telah dibangkitkan. Nilai *fitness* akan digunakan untuk menentukan baik tidaknya setiap individu. Proses perhitungan *fitness* dapat dilihat pada *Source Code 5.2*

```
1 public void kalkulasi(int bobot, float pakanharian)
2     {
3         this.total_pk_parent = 0;
4         this.total_tdn_parent = 0;
5         this.totalHarga = 0;
6         this.totalKromosom = 0;
7
8         for (int j = 0; j < getTotalBahan(); j++)
9             {
10                totalKromosom +=
11                getBahanPakan(j).getPanjangStringKromosom();
12            }
13
14        for (int i = 0; i < bahanPakanRow.Count; i++)
15            {
16                this.totalHarga +=
17                ((bahanPakanRow[i].getPanjangStringKromosom() / totalKromosom) *
18                pakanharian) * bahanPakanRow[i].harga;
19                this.total_pk_parent +=
20                (((bahanPakanRow[i].getPanjangStringKromosom() / totalKromosom) *
21                pakanharian) * (bahanPakanRow[i].pk) / 100);
22                this.total_tdn_parent +=
23                (((bahanPakanRow[i].getPanjangStringKromosom() / totalKromosom) *
24                pakanharian) * (bahanPakanRow[i].tdn) / 100);
25            }
26
27        tampilText += "Total Harga: <b>" + totalHarga + "</b>"
28        <br/>";
29
30        this.penalty_pk = checkZero(sapi.pk -
31        total_pk_parent);
32        this.penalty_tdn = checkZero(sapi.tdn -
33        total_tdn_parent);
34        this.fitness = checkZero(10000 / (totalHarga +
35        (penalty_pk + penalty_tdn)*5000));
36    }
37
38    public float checkZero(float value)
39    {
40        if (value < 0) return 0;
41    }
42    }
```

```

39         return value;
40     }}

```

### Source Code 5.2 Perhitungan *Fitness*

Penjelasan dari *source code* pada *Source Code 5.2* adalah sebagai berikut:

1. Baris 3 - 6 merupakan proses inialisasi variabel
2. Baris 8 – 24 merupakan proses perhitungan harga dan jumlah nutrisi. Baris 8 – 12 merupakan proses perhitungan total kromosom. Baris 14 – 18 merupakan proses perhitungan total harga masing-masing kromosom. Baris 19 - 24 merupakan proses perhitungan total nutrisi sapi potong setiap kromosom
3. Baris 26 - 27 merupakan proses untuk menampilkan total harga
4. Baris 28 – 31 merupakan proses perhitungan *penalty*
5. Baris 32 – 34 merupakan proses perhitungan nilai *fitness*
6. Baris 36 – 40 merupakan proses untuk mengecek nilai nol pada *penalty* maupun *fitness*

#### 5.1.3 Implementasi Proses untuk *Crossover*

Dalam proses *crossover* terjadi proses pemilihan induk. Metode *crossover* yang digunakan dalam kasus ini adalah *extended intermediate crossover* yang menghasilkan *offspring* dari kombinasi nilai dua induk. Proses perhitungan *crossover* dapat dilihat pada *Source Code 5.3*

```

1 private void crossover(bool tampilan)
2     {
3         if (offspring != null)
4             {
5                 offspring.Clear();
6                 offspring = null;
7                 System.GC.Collect();
8             }
9
10        int size = parent.getTotalBahan();
11        rdmAlfa = new float[size];
12
13        for (int j = 0; j < size; j++)
14            {
15                rdmAlfa[j] = addRandom();

```

```
16     }
17     offspring = new List<Parent>();
18     daftarParent = new List<string>();
19     childCr = Convert.ToInt32(cr * nParent);
20     for (int i = 0; i < childCr ; i++)
21     {
22         if (i == 0 || i % 2 == 0)
23         {
24             idx1 = NextInteger();
25             idx2 = NextInteger();
26
27             while (idx2 == idx1)
28                 idx2 = NextInteger();
29         }
30         Parent child = new Parent(sapi);
31         Parent parent1 = parents[idx1];
32         Parent parent2 = parents[idx2];
33         daftarParent.Add(parent1.nama + " & " +
34 parent2.nama);
35
36         for (int j = 0; j < size; j++)
37         {
38             BahanPakan pakanChild =
39 (BahanPakan)parent1.getBahanPakan(j).Clone();
40
41             if (i == 0 || i % 2 == 0)
42             {
43                 pakanChild.setPanjangStringKromosom (parent1.getBahanPakan(j).getPan
44 jangStringKromosom() + rdmAlfa[j] *
45 (parent2.getBahanPakan(j).getPanjangStringKromosom() -
46 parent1.getBahanPakan(j).getPanjangStringKromosom()));
47             }
48             else
49                 pakanChild.setPanjangStringKromosom (parent2.getBahanPakan(j).getPan
50 jangStringKromosom() + rdmAlfa[j] *
51 (parent1.getBahanPakan(j).getPanjangStringKromosom() -
52 parent2.getBahanPakan(j).getPanjangStringKromosom()));
53
54                 while (pakanChild.getPanjangStringKromosom() < 0)
55                 {
56                     pakanChild.setPanjangStringKromosom (NextFloat (random));
57                 }
58                 child.addBahanPakan (pakanChild);
59             }
60         }
61     }
```

62	<code>child.kalkulasi(bobot, pakanharian);</code>
63	<code>child.nama = "C" + i + "_" + iterasi;</code>
64	<code>offspring.Add(child);}}</code>

### Source Code 5.3 Perhitungan Crossover

Penjelasan dari *source code* pada *Source Code 5.3* adalah sebagai berikut:

1. Baris 3 – 8 merupakan proses untuk mengosongkan nilai offspring
2. Baris 10 – 11 merupakan proses inisialisasi variabel
3. Baris 13 – 16 merupakan proses pengulangan sebanyak ukuran “size” untuk mendapatkan nilai random alfa
4. Baris 17 – 19 merupakan proses inisialisasi variabel
5. Baris 20 – 29 merupakan proses random *parent*
6. Baris 30 – 32 merupakan proses inisialisasi variabel
7. Baris 33 merupakan proses pemberian nama asal *parent*
8. Baris 38 – 39 merupakan proses menyalin bahanpakan ke *pakanChild*
9. Baris 41 – 47 merupakan proses *crossover* untuk indeks yang genap
10. Baris 49 – 53 merupakan proses *crossover* untuk indeks yang ganjil
11. Baris 53 – 60 merupakan proses ulang random kromosom jika anak dari *crossover* menghasilkan nilai negatif
12. Baris 62 merupakan menghitung *nutrisi*, *harga*, *penalty*, dan *fitness*.
13. Baris 63 merupakan proses penamaan *child*
14. Baris 64 merupakan proses penambahan variabel *child* kedalam variabel *offspring*

#### 5.1.4 Implementasi Proses untuk Mutasi

Mutasi dilakukan dengan memilih satu induk secara acak dari populasi. Metode mutasi yang digunakan adalah random mutation yang dilakukan dengan menambah atau mengurangi nilai gen terpilih dengan bilangan random yang kecil. Proses perhitungan mutasi dapat dilihat pada *Source Code 5.4*

```
1 private void mutasi(bool tampilan)
2     {
3         daftarParent = new List<string>();
4         childMr = Convert.ToInt32(mr * nParent);
5
6         for (int i = 0; i < childMr; i++)
7             {
8                 idx1 = NextInteger();
9                 parentmut = parents[idx1];
10                int size = parentmut.getTotalBahan();
11                Parent childmut = new Parent(sapi);
12
13                daftarParent.Add(parentmut.nama);
14                int idxgen = RandomGen();
15
16                for (int j = 0; j < size; j++)
17                    {
18                        BahanPakan pakanChildmut =
19 (BahanPakan)parentmut.getBahanPakan(j).Clone();
20                        if (j == idxgen)
21                            {
22                                pakanChildmut.setPanjangStringKromosom(parentmut.getBahanPakan(idxgen).getPanjangStringKromosom() + (randomR() * (maxRandom - minRandom)));
23
24                                }
25                            else
26                                {
27                                    pakanChildmut.setPanjangStringKromosom(parentmut.getBahanPakan(j).getPanjangStringKromosom());
28                                }
29
30                                while (pakanChildmut.getPanjangStringKromosom() < 0)
31                                    {
32                                        pakanChildmut.setPanjangStringKromosom(NextFloat(random));
33                                    }
34                                childmut.addBahanPakan(pakanChildmut);
35                            }
36
37                childmut.kalkulasi(bobot, pakanharian);
38                childmut.nama = "C" + (i+childCr) + "_" + iterasi;
39                offspring.Add(childmut);
40            }
41    }
```

**Source Code 5.4** Perhitungan Mutasi

Penjelasan dari *source code* pada *Source Code 5.4* adalah sebagai berikut:

1. Baris 3 - 4 merupakan proses inisialisasi variabel
2. Baris 8 – 9 merupakan proses random *parent*
3. Baris 10 - 11 merupakan proses inisialisasi variabel
4. Baris 13 merupakan proses pemberian nama asal *parent*
5. Baris 14 merupakan proses random gen
6. Baris 18 – 19 merupakan proses menyalin bahanpakan ke pakanChildmut
7. Baris 20 – 25 merupakan proses mutasi jika indeks kromsوم yang akan dimutasi sesuai dengan indeks hasil random gen
8. Baris 27 - 31 merupakan proses pengambilan nilai gen kromosوم yang tidak terkena mutasi
9. Baris 33 – 38 merupakan proses ulang random kromosوم jika anak dari *crossover* menghasilkan nilai negatif
10. Baris 40 merupakan menghitung nutrisi, harga, *penalty*, dan *fitness*
11. Baris 41 merupakan proses penamaan childmut
12. Baris 42 merupakan proses penambahan variabel childmut kedalam variabel offspring

### 5.1.5 Implementasi Proses untuk Seleksi

Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan pada generasi berikutnya (Mahmudy, 2013). Fungsi probabilitas digunakan untuk memilih individu yang dipertahankan hidup. Proses perhitungan seleksi dapat dilihat pada *Source Code 5.5*

```

1 private void seleksi(bool tampilan)
2 {
3     parents.AddRange(offspring);
4     parents.Sort(sorter);
5     for (int i = 0; i < (childCr + childMr) ; i++)
6         {
7             parents.RemoveAt(parents.Count - 1);
8         }
9     }

```

**Source Code 5.5** Perhitungan Seleksi

Penjelasan dari *source code* pada *Source Code* 5.5 adalah sebagai berikut:

1. Baris 3 merupakan proses penambahan list offspring kedalam list parents
2. Baris 5 merupakan proses *sorting* yang akan mengurutkan nilai fitness terbesar hingga *fitness* terkecil
3. Baris 7 - 10 merupakan proses untuk menampilkan hasil *sorting* sejumlah *parent* awal

## 5.2 Implementasi Antarmuka Aplikasi

Antarmuka aplikasi sistem “Optimasi Komposisi Pakan Sapi Potong Menggunakan Algoritma Genetika” nantinya akan digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Antarmuka perangkat lunak terdiri dari tiga halaman, yaitu halaman utama, halaman bahan pakan dan halaman parameter. Berikut hasil implementasi antarmuka pada masing-masing halaman pada aplikasi ini.

### 5.2.1 Tampilan Halaman Utama

Halaman ini merupakan halaman awal ketika pengguna masuk kedalam aplikasi. Halaman ini merupakan halaman utama yang digunakan pengguna untuk memasukkan data yang dibutuhkan oleh sistem, seperti data sapi, data paramater, dan data bahan pakan. Sistem akan mengolah input tersebut menjadi keluaran yang ditampilkan di detail perhitungan, kromosom terbaik, dan rekomendasi bahan pakan yang ditawarkan oleh sistem.

Ransum Sapi Potong Algoritma Genetika

INPUT DATA SAPI POTONG

Parameter Setting

Jenis Sapi: Sapi Jantan

Bobot Badan: 250

Pertambahan Bobot Badan: 1

INPUT DATA BAHAN PAKAN

Masukkan bahan pakan yang tersedia

ID	Bahan Pakan
1	Dedak Padi Kasar
2	Dedak Padi Halus
3	Dedak Terigu Halus

>> HITUNG <<

**KROMOSOM TERBAIK**

**Iterasi ke-1**

Fitness terbaik: 0.2496106

Parent: P4

Kromosom terbaik: 3.032363| 1.458978| 7.259789|

**Iterasi ke-2**

Fitness terbaik: 0.2496106

Parent: P4

Kromosom terbaik: 3.032363| 1.458978| 7.259789|

**Iterasi ke-3**

Fitness terbaik: 0.2496106

Parent: P4

Kromosom terbaik: 3.032363| 1.458978| 7.259789|

**Iterasi ke-4**

Fitness terbaik: 0.2496106

Parent: P4

Kromosom terbaik: 3.032363| 1.458978| 7.259789|

**DETAIL PERHITUNGAN**

Nutrisi yang dibutuhkan sapi  
Protein kasar : 0.76  
Energi / TDN : 4.3

**PARENT AWAL**

Parent	Dedak Padi Kasar			Dedak Padi Halus			Dedak Terigu Halus			Total Kromosom	Bobot Sapi	Pakan Harian	Total PK	Total TDN	Penalty PK	Penalty TDN	Total Harga	Fitness
	Kromosom	PK (%)	TDN (%)	Kromosom	PK (%)	TDN (%)	Kromosom	PK (%)	TDN (%)									
P1	5.967851	13.8	5.5	3.188752	15.9	6.7	1.279601	18.9	7.0	10.4362	250	7.5	1.130023	2.41497	0	1.88503	39791.6	0.2031828
P2	8.67045	13.8	5.5	2.296117	15.9	6.7	6.28884	18.9	7.0	17.25541	250	7.5	1.195362	2.789326	0	1.510674	38498	0.2171488
P3	6.830877	13.8	5.5	3.255956	15.9	6.7	9.917987	18.9	7.0	20.00482	250	7.5	1.25027	3.561559	0	0.7384412	38720.69	0.2357773
P4	3.032363	13.8	5.5	1.458978	15.9	6.7	7.259789	18.9	7.0	11.75113	250	7.5	1.290861	3.973754	0	0.3262458	38431.17	0.2496106
P5	9.116837	13.8	5.5	0.8058005	15.9	6.7	4.687541	18.9	7.0	14.61018	250	7.5	1.166408	2.218962	0	2.081038	37913.65	0.2069586
P6	5.610626	13.8	5.5	0.1273395	15.9	6.7	8.308483	18.9	7.0	14.04645	250	7.5	1.262677	3.3157	0	0.9843006	37567.99	0.2353523
P7	9.46135	13.8	5.5	3.078258	15.9	6.7	7.676739	18.9	7.0	20.21635	250	7.5	1.204228	2.951766	0	1.348234	38642	0.220346
P8	2.794598	13.8	5.5	4.006691	15.9	6.7	2.526208	18.9	7.0	9.327497	250	7.5	1.206249	3.703993	0	0.5960071	40721.68	0.228824
P9	5.902109	13.8	5.5	1.224722	15.9	6.7	5.891219	18.9	7.0	13.01805	250	7.5	1.222915	3.035612	0	1.264388	38205.59	0.2245802
P10	2.78716	13.8	5.5	1.16666	15.9	6.7	0.4634913	18.9	7.0	4.417311	250	7.5	1.116732	2.138291	0	2.161709	39480.83	0.1988492

**REKOMENDASI**

Jumlah berat pakan yang diperlukan setiap hari: 7.5kg

Jumlah Dedak Padi Kasar: 1.27451717853546 kg dengan harga Rp. 6373  
 Jumlah Dedak Padi Halus: 0.62331634759903 kg dengan harga Rp. 3740  
 Jumlah Dedak Terigu Halus: 5.60216665267944 kg dengan harga Rp. 28011

Total harga bahan: Rp. 38124

Gambar 5.1 Halaman Utama

Halaman utama terdiri dari beberapa tampilan, yaitu tampilan detail perhitungan yang merupakan bagian dari halaman awal yang menampilkan detail hasil perhitungan. Detail hasil perhitungan yang diperlihatkan adalah detail perhitungan pada beberapa generasi awal dan generasi terakhir. Tampilan detail perhitungan ditunjukkan pada Gambar 5.2.

**DETAIL PERHITUNGAN**

Nutrisi yang dibutuhkan sapi  
Protein kasar : 0.76  
Energi / TDN : 4.3

**PARENT AWAL**

Parent	Dedak Padi Kasar			Dedak Padi Halus			Dedak Terigu Halus			Total Kromosom	Bobot Sapi	Pakan Harian	Total PK	Total TDN	Penalty PK	Penalty TDN	Total Harga	Fitness
	Kromosom	PK (%)	TDN (%)	Kromosom	PK (%)	TDN (%)	Kromosom	PK (%)	TDN (%)									
P1	5.967851	13.8	5.5	3.188752	15.9	6.7	1.279601	18.9	7.0	10.4362	250	7.5	1.130023	2.41497	0	1.88503	39791.6	0.2031828
P2	8.67045	13.8	5.5	2.296117	15.9	6.7	6.28884	18.9	7.0	17.25541	250	7.5	1.195362	2.789326	0	1.510674	38498	0.2171488
P3	6.830877	13.8	5.5	3.255956	15.9	6.7	9.917987	18.9	7.0	20.00482	250	7.5	1.25027	3.561559	0	0.7384412	38720.69	0.2357773
P4	3.032363	13.8	5.5	1.458978	15.9	6.7	7.259789	18.9	7.0	11.75113	250	7.5	1.290861	3.973754	0	0.3262458	38431.17	0.2496106
P5	9.116837	13.8	5.5	0.8058005	15.9	6.7	4.687541	18.9	7.0	14.61018	250	7.5	1.166408	2.218962	0	2.081038	37913.65	0.2069586
P6	5.610626	13.8	5.5	0.1273395	15.9	6.7	8.308483	18.9	7.0	14.04645	250	7.5	1.262677	3.3157	0	0.9843006	37567.99	0.2353523
P7	9.46135	13.8	5.5	3.078258	15.9	6.7	7.676739	18.9	7.0	20.21635	250	7.5	1.204228	2.951766	0	1.348234	38642	0.220346
P8	2.794598	13.8	5.5	4.006691	15.9	6.7	2.526208	18.9	7.0	9.327497	250	7.5	1.206249	3.703993	0	0.5960071	40721.68	0.228824
P9	5.902109	13.8	5.5	1.224722	15.9	6.7	5.891219	18.9	7.0	13.01805	250	7.5	1.222915	3.035612	0	1.264388	38205.59	0.2245802
P10	2.78716	13.8	5.5	1.16666	15.9	6.7	0.4634913	18.9	7.0	4.417311	250	7.5	1.116732	2.138291	0	2.161709	39480.83	0.1988492

Gambar 5.2 Detail Perhitungan

Tampilan selanjutnya adalah tampilan kromosom terbaik yang menampilkan bobot dan harga masing-masing bahan pakan yang dibutuhkan sapi tiap harinya dan ditunjukkan pada Gambar 5.3

REKOMENDASI	
Jumlah berat pakan yang diperlukan setiap hari: 7.5kg	
Jumlah Dedak Padi Kasar: 1.27451717853546 kg dengan harga Rp. 6373 Jumlah Dedak Padi Halus: 0.62331634759903 kg dengan harga Rp. 3740 Jumlah Dedak Terigu Halus: 5.60216665267944 kg dengan harga Rp. 28011	
Total harga bahan: Rp. 38124	

**Gambar 5.3** Kromosom Terbaik

Tampilan terakhir adalah tampilan kromosom terbaik yang menampilkan meliputi kromosom terbaik, fitness, dan parent asal kromosom tersebut dan ditunjukkan pada Gambar 5.4.

KROMOSOM TERBAIK	
<b>Iterasi ke-1</b>	
Fitness terbaik: 0.2496106 Parent: P4 Kromosom terbaik: 3.032363  1.458978  7.259789	
<b>Iterasi ke-2</b>	
Fitness terbaik: 0.2496106 Parent: P4 Kromosom terbaik: 3.032363  1.458978  7.259789	
<b>Iterasi ke-3</b>	
Fitness terbaik: 0.2496106 Parent: P4 Kromosom terbaik: 3.032363  1.458978  7.259789	
<b>Iterasi ke-4</b>	
Fitness terbaik: 0.2496106 Parent: P4 Kromosom terbaik:	

**Gambar 5.4** Kromosom Terbaik

### 5.2.2 Tampilan Halaman Bahan Pakan

Halaman ini merupakan halaman yang digunakan pengguna untuk menginputkan bahan pakan yang tersedia.

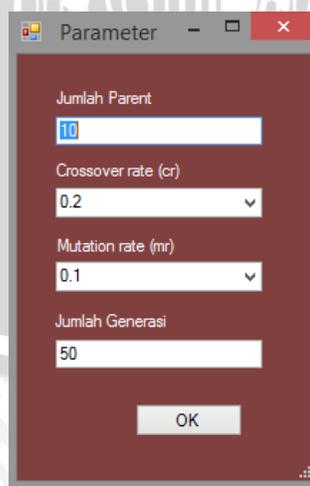


ID	Bahan Pakan	Protein Kasar	Energi / TDN	Harga
<input checked="" type="checkbox"/>	1 Dedak Padi Kasar	13.8	5.5	5000
<input checked="" type="checkbox"/>	2 Dedak Padi Halus	15.9	67	6000
<input checked="" type="checkbox"/>	3 Dedak Terigu Halus	18.9	70	5000
<input type="checkbox"/>	4 Polard	17	70	4000
<input type="checkbox"/>	5 Tepung Jagung Kuning	10.8	90	4500
<input type="checkbox"/>	6 Tepung Gaplek	2.3	78	3500
<input type="checkbox"/>	7 Tepung Ikan	49	59	4000
<input type="checkbox"/>	8 Ampas Tahu	23.7	79	3500
<input type="checkbox"/>	9 Ampas Bir	33.7	74	3000
<input type="checkbox"/>	10 Ampas Kecap	23.5	87	3500
<input type="checkbox"/>	11 Bungkil Kelapa	21.2	81	4000
<input type="checkbox"/>	12 Bungkil Kelapa Sawit	16.5	70	5500
<input type="checkbox"/>	13 Bungkil Kacang Tanah	33.7	81	5000
<input type="checkbox"/>	14 Tepung Biji Kapuk	32.7	74	4000
<input type="checkbox"/>	15 Bungkil Kedelai	41.3	83.2	3000

Gambar 5.5 Halaman Bahan Pakan

### 5.2.3 Tampilan Halaman Parameter

Halaman ini merupakan halaman yang digunakan pengguna untuk menentukan parameter yang digunakan untuk perhitungan algoritma genetika.



Parameter

Jumlah Parent  
10

Crossover rate (cr)  
0.2

Mutation rate (mr)  
0.1

Jumlah Generasi  
50

OK

Gambar 5.6 Halaman Parameter

## BAB VI

### PENGUJIAN DAN ANALISIS

Bab ini akan membahas mengenai pengujian dan analisis terhadap sistem yang telah diimplementasikan pada tahap sebelumnya. Bab ini berisi tiga sub bab, yaitu sistematika pengujian, analisis dan pembahasan, serta solusi terbaik yang pernah didapatkan.

#### 6.1 Sistematika Pengujian

Terdapat tiga pengujian yang dilakukan pada penelitian ini. Pada setiap pengujian, akan dilihat hasil *fitness* paling optimal, sehingga dapat ditemukan parameter terbaik yang digunakan untuk menyelesaikan permasalahan optimasi komposisi pakan ternak sapi potong. Pengujian ini dilakukan dengan masukan sebagai berikut:

a. Sapi potong

Jenis : Sapi Jantan

Bobot : 250 kg

Pertambahan bobot : 1 kg

b. Bahan Pakan

Terdapat lima bahan pakan yang digunakan, yaitu: dedak padi halus, dedak terigu halus, tepung jagung kuning, tepung ikan, dan bungkil kedelai.

#### 6.2 Analisis dan Pembahasan

Proses pengujian dilakukan sesuai perancangan uji coba yang telah dijelaskan pada Bab 4. Tiga pengujian yang dilakukan pada penelitian ini yaitu pengujian ukuran populasi, banyaknya generasi, dan kombinasi *cr* dan *mr*. Data yang digunakan pada pengujian ini adalah data bahan pakan yang ada pada Tabel 2.3 dan data sapi potong pada Tabel 2.4, Tabel 2.5, dan Tabel 2.6.

### 6.2.1 Pengujian dan Analisis Ukuran Populasi

Pengujian ukuran populasi digunakan untuk menentukan ukuran populasi yang terbaik untuk menghasilkan solusi terbaik dalam kasus ini. Ukuran populasi yang akan diuji adalah 50, 100, 150, 200, 250, 300, 350, 400, dan 450. *Crossover rate* dan *mutation rate* yang akan diuji adalah 0.5 dan 0.1 yang didapatkan dari hasil parameter terbaik dari penelitian sebelumnya oleh Ariwibowo (2011) yang berjudul “Penerapan Algoritma Genetika Pada Penentuan Komposisi Pakan Ayam Petelur”. Untuk lebih detailnya mengenai parameter yang digunakan pada uji coba populasi adalah sebagai berikut :

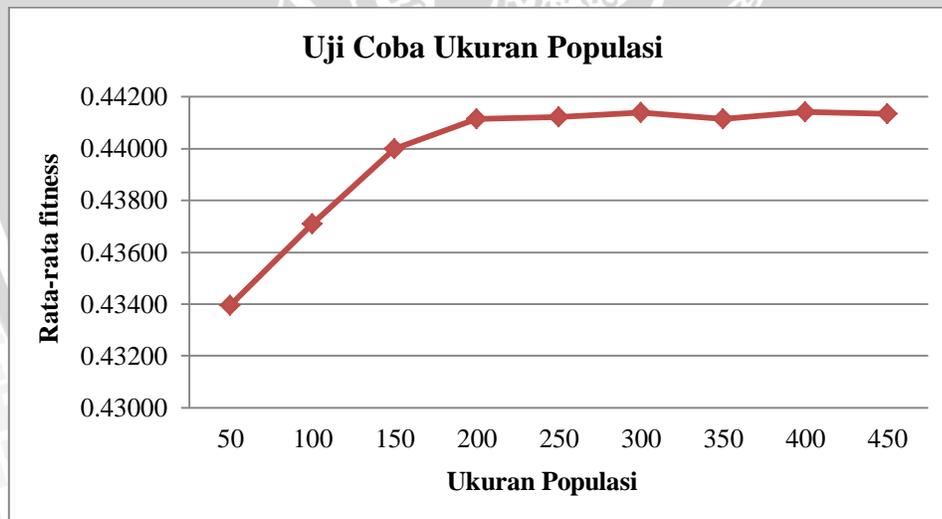
- a. Ukuran populasi = 50 – 450
- b. Banyaknya generasi = 100
- c. *Crossover Rate* = 0.5
- d. *Mutation Rate* = 0.1

Pengujian dilakukan sepuluh kali untuk mendapatkan hasil yang mewakili kemampuan algoritma secara utuh. *Fitness* terbaik pada setiap percobaan dihitung rata-ratanya untuk mengetahui ukuran populasi paling optimal. Hasil pengujian menunjukkan semakin besar ukuran populasi maka *fitness* yang dihasilkan cenderung semakin baik. Hasil uji coba ukuran populasi dapat dilihat pada Tabel 6.1.

**Tabel 6.1** Hasil Uji Coba Ukuran Populasi

Ukuran Populasi	Nilai Fitness										Rata – rata <i>fitness</i>
	Pengujian ke -										
	1	2	3	4	5	6	7	8	9	10	
50	0.43974	0.43925	0.43668	0.43330	0.42751	0.43699	0.43928	0.41422	0.43666	0.43595	0.43396
100	0.43271	0.42860	0.43757	0.43917	0.44106	0.44230	0.43167	0.43902	0.43817	0.44078	0.43710
150	0.43603	0.43939	0.43936	0.43823	0.44072	0.44138	0.44137	0.44028	0.44140	0.44171	0.43999
200	0.44046	0.44150	0.44259	0.44075	0.44278	0.44225	0.44126	0.43730	0.44285	0.43978	0.44115
250	0.44141	0.44178	0.43828	0.44197	0.44225	0.43871	0.44193	0.44304	0.44186	0.44088	0.44121
300	0.44172	0.44068	0.44044	0.44203	0.44190	0.44136	0.44180	0.44091	0.44110	0.44205	0.44140
350	0.44279	0.44090	0.44318	0.44244	0.43816	0.44119	0.43977	0.44136	0.44169	0.44012	0.44116
400	0.44027	0.44111	0.44133	0.44152	0.44147	0.44179	0.44167	0.44197	0.44153	0.44144	0.44141
450	0.44269	0.44197	0.44206	0.44270	0.44120	0.44012	0.44265	0.44208	0.43586	0.44223	0.44135

Berdasarkan grafik hasil uji coba pada Gambar 6.1, ditunjukkan bahwa semakin besar ukuran populasi, maka rata-rata *fitness* yang dihasilkan cenderung meningkat. Pada grafik tersebut, dapat dilihat rata-rata *fitness* dari ukuran populasi 150 menuju ukuran populasi 200 meningkat. Rata-rata *fitness* yang dihasilkan ukuran populasi diatas 200 cenderung stabil. Hal ini menunjukkan bahwa ukuran populasi 200 adalah ukuran populasi yang paling optimal. Perubahan yang tidak begitu besar ini terjadi karena anak yang dihasilkan pada proses reproduksi mirip dengan induknya (Mahmudy, 2013). Pola peningkatan nilai *fitness* yang sebanding dengan peningkatan ukuran populasi ini juga diperoleh oleh Pratiwi, Mahmudy, dan Dewi (2014) yang menerapkan algoritma genetika untuk mengoptimasi biaya pemenuhan kebutuhan gizi. Pola yang sama juga ditemukan pada penelitian lain yang menerapkan algoritma genetika untuk *vehicle routing problem with time window (VRPTW)* pada kasus optimasi beras bersubsidi yang dilakukan oleh Putri, Mahmudy, dan Ratnawati (2014).



**Gambar 6.1** Grafik Hasil Uji Coba Ukuran Populasi

### 6.2.2 Pengujian dan Analisis Banyaknya Generasi

Pengujian banyaknya generasi dilakukan untuk menentukan banyaknya generasi yang dapat menghasilkan solusi terbaik dalam kasus ini. Pada pengujian

banyaknya generasi ini, digunakan ukuran populasi hasil uji coba sebelumnya yaitu ukuran populasi 200 yang dianggap dapat menghasilkan rata – rata nilai fitness paling optimal. Banyaknya generasi yang diuji adalah 50, 75, 100, 125, 150, 175, 200, 225 dan 250. Untuk lebih detailnya mengenai parameter yang digunakan pada uji coba banyaknya generasi adalah sebagai berikut :

- a. Ukuran populasi = 200
- b. Banyaknya generasi = 50 - 250
- c. *Crossover Rate* = 0.5
- d. *Mutation Rate* = 0.1

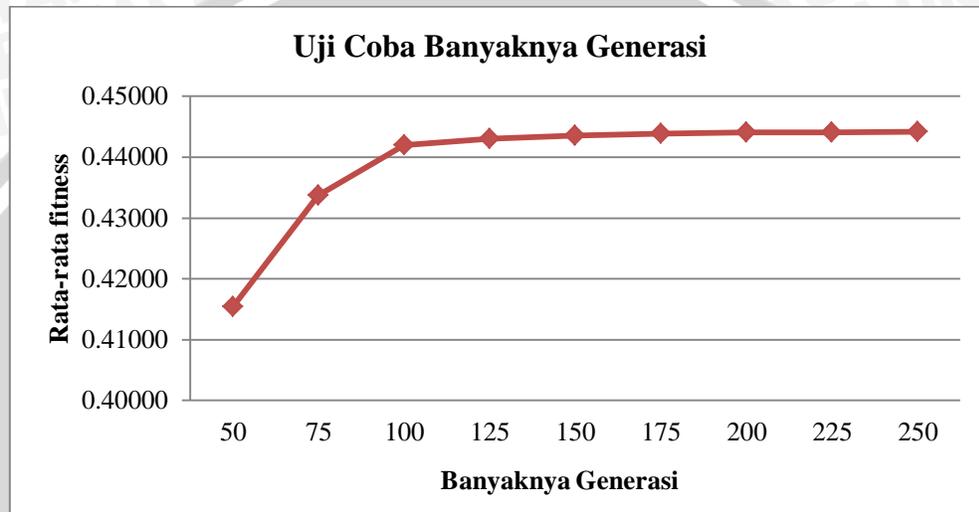
Pengujian banyaknya generasi ini dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 6.2.

**Tabel 6.2** Hasil Uji Coba Banyaknya generasi

Gene-Rasi	Nilai Fitness										Rata – rata fitness
	Pengujian ke -										
	1	2	3	4	5	6	7	8	9	10	
50	0.41434	0.42492	0.39413	0.42996	0.39980	0.42879	0.40275	0.40834	0.41942	0.43133	0.41538
75	0.43095	0.43421	0.43802	0.43431	0.43877	0.43155	0.43630	0.42638	0.43059	0.43642	0.43375
100	0.44156	0.44225	0.44142	0.44187	0.44221	0.44208	0.44249	0.44205	0.44163	0.44196	0.44195
125	0.44324	0.44291	0.44296	0.44265	0.44258	0.44320	0.44244	0.44323	0.44348	0.44329	0.44300
150	0.44358	0.44373	0.44392	0.44397	0.44374	0.44385	0.44305	0.44303	0.44283	0.44372	0.44354
175	0.44401	0.44385	0.44388	0.44384	0.44415	0.44382	0.44400	0.44364	0.44352	0.44409	0.44388
200	0.44385	0.44408	0.44400	0.44420	0.44431	0.44407	0.44412	0.44406	0.44430	0.44418	0.44412
225	0.44397	0.44401	0.44424	0.44410	0.44430	0.44397	0.44416	0.44410	0.44400	0.44378	0.44406
250	0.44429	0.44429	0.44404	0.44422	0.44426	0.44423	0.44427	0.44416	0.44425	0.44414	0.44422

Berdasarkan grafik hasil uji coba pada Gambar 6.2, dapat dilihat rata-rata *fitness* mengalami peningkatan dari generasi 75 menuju generasi 100. Rata-rata *fitness* yang dihasilkan generasi diatas 100 cenderung stabil karena perubahan rata-rata *fitness* yang tidak begitu besar. Hal ini menunjukkan bahwa generasi sebanyak 100 adalah banyaknya generasi yang paling optimal. Semakin banyak generasi maka semakin besar waktu komputasi, namun belum tentu menghasilkan solusi yang lebih baik (Mahmudy, 2013). Pola seperti ini juga ditemukan dua penelitian lain yaitu penelitian pertama yang menerapkan algoritma genetika untuk mengoptimasi biaya

pemenuhan kebutuhan gizi oleh Pratiwi, Mahmudy, dan Dewi (2014) dan penelitian lain oleh Putri, Mahmudy, dan Ratnawati (2014) yang menerapkan algoritma genetika untuk *vehicle routing problem with time window (VRPTW)* pada kasus optimasi beras bersubsidi.



**Gambar 6.2** Grafik Hasil Uji Coba Banyaknya generasi

### 6.2.3 Pengujian dan Analisis Kombinasi *Cr* dan *Mr*

Pengujian *crossover rate (cr)* dan *mutation rate (mr)* dilakukan untuk mengetahui kombinasi *cr* dan *mr* terbaik yang dapat menghasilkan fitness paling optimal. Nilai *cr* dan *mr* yang digunakan berkisar antara 0 dan 0.6 dan keduanya jika dijumlahkan menghasilkan nilai 0.6. Pengujian *cr* dan *mr* juga menggunakan hasil pengujian sebelumnya yaitu hasil pengujian ukuran populasi dan pengujian banyaknya generasi yang menghasilkan nilai *fitness* paling optimal. Untuk lebih detailnya mengenai parameter yang digunakan dalam uji coba kombinasi *cr* dan *mr* adalah sebagai berikut :

- Ukuran populasi = 200
- Banyaknya generasi = 100

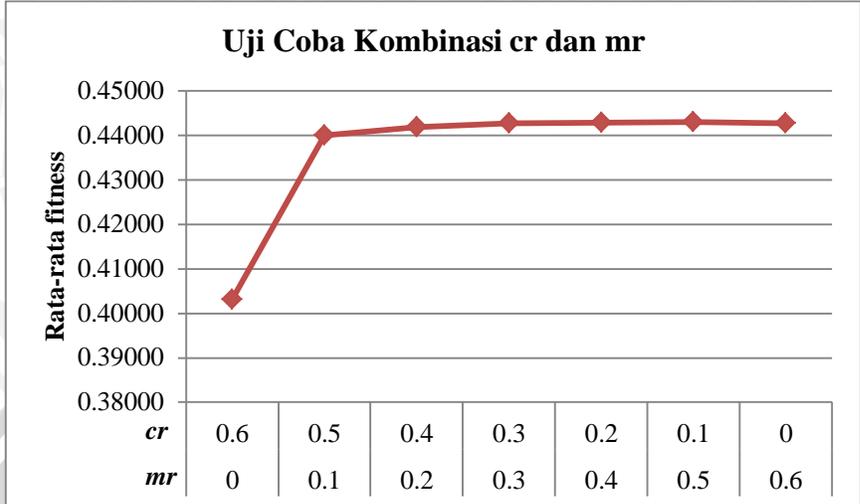
Pengujian ukuran populasi ini dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Uji Coba Kombinasi  $Cr$  dan  $Mr$ 

$Cr$	$Mr$	Nilai Fitness										Rata – rata <i>fitness</i>
		Pengujian ke -										
		1	2	3	4	5	6	7	8	9	10	
0.6	0	0.40171	0.39125	0.39225	0.41860	0.41076	0.39668	0.41561	0.40888	0.41319	0.38272	0.40317
0.5	0.1	0.44140	0.44135	0.44056	0.44076	0.44123	0.43959	0.43814	0.43815	0.43628	0.44316	0.44006
0.4	0.2	0.43963	0.44123	0.44241	0.44204	0.44320	0.44223	0.44177	0.44339	0.43975	0.44297	0.44186
0.3	0.3	0.44241	0.44313	0.44314	0.44241	0.44260	0.44308	0.44243	0.44334	0.44285	0.44239	0.44278
0.2	0.4	0.44361	0.44324	0.44297	0.44356	0.44276	0.44264	0.44297	0.44307	0.44275	0.44162	0.44292
0.1	0.5	0.44296	0.44238	0.44285	0.44203	0.44367	0.44347	0.44227	0.44273	0.44391	0.44359	0.44298
0	0.6	0.44290	0.44282	0.44368	0.44357	0.44205	0.44353	0.44223	0.44107	0.44352	0.44283	0.44282

Berdasarkan grafik hasil uji coba pada Gambar 6.3, rata-rata *fitness* yang dihasilkan sangat beragam. Pada grafik tersebut dapat dilihat bahwa rata-rata *fitness* yang dihasilkan dari kombinasi  $cr$  0.6 dan  $mr$  0 menuju ke kombinasi  $cr$  0.5 dan  $mr$  0.1 meningkat tajam. Rata-rata *fitness* yang dihasilkan setelah kombinasi itu cenderung stabil karena perubahan rata-rata *fitness* yang tidak begitu besar. Maka dari itu, dapat disimpulkan bahwa kombinasi  $cr$  0.5 dan  $mr$  0.1 merupakan nilai kombinasi yang paling optimal.

Permasalahan yang ingin diselesaikan dipengaruhi oleh kombinasi nilai parameter yang tepat (Mahmudy, 2013). Kombinasi  $cr$  dan  $mr$  yang dihasilkan pada setiap kasus akan menunjukkan hasil yang berbeda tergantung permasalahan yang akan diselesaikan. Hal ini disebabkan karena tidak ada suatu ketetapan nilai kombinasi  $cr$  dan  $mr$  yang dapat digunakan untuk menghasilkan solusi optimal.



Gambar 6.3 Grafik Hasil Uji Coba Kombinasi Cr dan Mr

### 6.3 Solusi Terbaik yang Pernah Didapatkan

Dalam penelitian optimasi menggunakan algoritma genetika, serangkaian pengujian pendahuluan diperlukan untuk mendapatkan kombinasi nilai parameter yang sesuai (Mahmudy, 2013). Pada penelitian ini, didapatkan beberapa parameter terbaik dengan rata-rata fitness paling optimal, yaitu ukuran populasi = 200, banyaknya generasi = 100, *crossover rate* = 0.5, dan *mutation rate* = 0.1. Nilai *fitness* yang dihasilkan dari parameter tersebut adalah 0.441545. Jika masukan bahan pakan sesuai dengan bahan pakan saat pengujian, yaitu dedak padi halus, dedak terigu halus, tepung jagung kuning, tepung ikan, dan bungkil kedelai akan menghasilkan rekomendasi pakan dedak padi halus dengan bobot 0.017583 kg seharga Rp.105, dedak terigu halus dengan bobot 0.0097326 seharga Rp.49, tepung jagung kuning dengan bobot 0.0430112 seharga Rp.194, tepung ikan dengan bobot 0.0110165 seharga Rp. 44, dan bungkil kedelai dengan bobot 7.4186563 kg seharga Rp.22256.

Setelah melakukan serangkaian uji coba, algoritma genetika dianggap mampu untuk menyelesaikan masalah optimasi komposisi pakan ternak sapi potong. Peternak dapat menggunakan rekomendasi komposisi pakan ternak yang telah ditemukan sehingga dapat memenuhi nutrisi ternak dan dapat meminimumkan biaya pengeluaran bahan pakan ternak.

## BAB VII PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil uji coba parameter algoritma genetika pada permasalahan optimasi pakan sapi potong, terdapat beberapa kesimpulan yaitu :

1. Algoritma genetika dapat menyelesaikan permasalahan optimasi pakan sapi potong. Hal ini dapat dilihat dari sistem yang dapat memenuhi kebutuhan nutrisi dan meminimalkan biaya bahan pakan sapi potong.
2. Parameter terbaik dengan rata – rata nilai fitness paling optimal yang didapatkan dari hasil pengujian adalah sebagai berikut :
  - Jumlah populasi : 200
  - Banyaknya generasi : 100
  - *Crossover Rate* : 0.5
  - *Mutation rate* : 0.1
3. Pengukuran solusi dari permasalahan optimasi pakan sapi potong ini dilakukan dengan perhitungan nilai fitness yang diperoleh dari total biaya pakan dan nilai *pinalty* yaitu nilai yang didapatkan dari kekurangan nutrisi sapi potong.

### 7.2 Saran

Penelitian ini dapat dikembangkan untuk menyelesaikan masalah optimasi pakan sapi potong dengan menambah kandungan nutrisi sapi potong yang lain selain protein kasar dan energi, menambahkan data pakan yang lebih banyak dan lebih bervariasi. Selain itu penggunaan metode *crossover*, mutasi dan seleksi yang berbeda juga dapat mempengaruhi menghasilkan hasil nilai *fitness* yang berbeda pada tiap individu.

## DAFTAR PUSTAKA

- Abidin, Zainal. 2002. *Penggemukan Sapi Potong*. PT. Agromedia Pustaka: Jakarta.
- Ariwibowo, Arnold. dkk. 2011. *Penerapan Algoritma Genetika Pada Penentuan Komposisi Pakan Ayam Petelur*. Universitas Pelita Harapan: Indonesia.
- Haryanti, Nina Woro. 2009. *Kualitas dan Kecukupan Nutrisi Sapi Sentimental di Peternakan Mitra Tani Andini, Kelurahan Gunung Pati, Kota Semarang*. Universitas Diponegoro: Semarang.
- Haryanto, Budi. 2012. *Perkembangan Penelitian Nutrisi Ruminansia*. Balai Penelitian Ternak: Bogor.
- Imbar VR & Jayanti 2011, 'Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Dengan Studi Kasus Pada Smp X ', Seminar Teknik Informatika dan Sistem Informasi, Fakultas Teknologi Informasi, Universitas Kristen Maranatha, Bandung.
- Mahmudy, Wayan Firdaus. 2006. *Penerapan Algoritma Genetika Pada Optimasi Model Penugasan*. Jurusan Matematika, FMIPA, Universitas Brawijaya: Malang.
- Mahmudy, Wayan Firdaus. 2013. *Algoritma Evolusi*. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya: Malang.
- Mathis, Clay P. 2003. *Protein and Energy Supplementation to Beef Cow Grazing New Mesico Rangelands*. New Mexico State University: Mexico.
- Ngadiyono, Nono. 2007. *Beternak Sapi*. PT. Citra Aji Pratama: Yogyakarta.
- Ngadiyono, Nono. 2012. *Beternak Sapi Potong Ramah Lingkungan*. PT. Citra Aji Pratama: Yogyakarta.
- Nugraha, Romada Andi. 2011. *Optimalisasi Formulasi Pakan Ternak Terhadap Ayam Pedaging dengan Menggunakan Metode Linear Programming*. Fakultas Teknologi Industri. Universitas Gunadharma: Jakarta.
- Parakkasi, Aminuddin. 1999. *Ilmu Nutrisi dan Makanan Ternak Ruminan*. Universitas Indonesia: Jakarta.

- Pratiwi, MI, Mahmudy, WF & Dewi, C. 2014. *Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 4, no. 6.
- Putri, FB, Mahmudy, WF & Ratnawati, DE. 2014. *Penerapan Algoritma Genetika Untuk Vehicle Routing Problem With Time Windows (VRPTW) Pada Kasus Optimasi Distribusi Beras Bersubsidi*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 1.
- Santosa, Undang. 2006. *Manajemen Usaha Ternak Potong*. Penebar Swadaya: Jakarta.
- Siregar, Sori Basya. 2008. *Penggemukan Sapi*. Penebar Swadaya: Jakarta.
- Soeprapto, Herry dan Abidin, Zainal. 2006. *Cara Tepat Penggemukkan Sapi Potong*. PT. Agromedia Pustaka: Depok.
- Sudarmono, A.S dan Sugeng, Bambang. 2008. *Sapi Potong*. Penebar Swadaya: Jakarta.
- Tim Laboratorium. 2012. *Pengetahuan Bahan Makanan Ternak*. Ilmu dan Teknologi Pakan Fakultas Peternakan IPB: Bogor.
- Wardhani, Luh Kesuma. dkk. 2011. *Optimasi Komposisi Bahan Pakan Ikan Air Tawar Menggunakan Metode Multi-Objective Genetic Algorithm*. Fakultas Sains dan Teknologi. UIN Sultan Syarif Kasim: Riau.

## LAMPIRAN

## Daftar Harga Bahan Pakan

No	Bahan Pakan	Harga (per kg)
1	Dedak padi kasar	5000
2	Dedak padi halus	6000
3	Dedak terigu kasar	4000
4	Dedak terigu halus	5000
5	Polard	4000
6	Tepung jagung kuning	4500
7	Tepung gaplek	3500
8	Tepung ikan	4000
9	Tepung terigu	5000
10	Ampas tahu	3500
11	Ampas bir	3000
12	Ampas kecap	3500
13	Bulgur	3500
14	Bungkil kelapa	4000
15	Bungkil kelapa sawit	5500
16	Bungkil kacang tanah	5000
17	Tepung biji kapuk	4000
18	Tepung darah	4500
19	Bungkil kedelai	3000
20	Tetes/molase	4000
21	Onggok	4000
22	Dedak merah	4000
23	Dedak kuning	5500
24	Dedak jagung	5000
25	Ampas sagu	4000
26	Bungkil geblek	5000
27	Bungkil arga	3500
28	Tepung biji kapas	4000
29	Kulit buah kakao	5000
30	Kecipir	4000
31	Bungkil wijen	5500

32	Singkong	5000
33	Ubi jalar	4000
34	Rumput lapangan	5000
35	Rumput serawit	3500
36	Rumput kampai	3600
37	Rumput molases	3000
38	Rumput gajah	3100
39	Rumput raja	3400
40	Rumpur setaria	3500
41	Daun jagung	3500
42	Daun kacang kedelai	3000
43	Daun kacang tanah	3500
44	Daun kol luar	3500
45	Daun lamtoro	3500
46	Pucuk tebu	3600
47	Daun turi	3000
48	Daun singkong	3100
49	Daun pisang	3400
50	Jerami padi	4000
51	Kudzu	5000
52	Batang pisang	3500