

repository.ub.ac.id

OPTIMASI KOMPOSISI PAKAN SAPI PERAH MENGGUNAKAN ALGORITMA GENETIKA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dewi Maida Safitri
NIM: 125150200111050



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016



PENGESAHAN

OPTIMASI KOMPOSISI PAKAN SAPI PERAH MENGGUNAKAN ALGORITMA
GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Dewi Maida Safitri

NIM: 125150200111050

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juni 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si., M.Kom

NIK: 201201 850719 1 001

Edy Santoso, S.Si., M.Kom

NIP: 19740414 200312 1 004

Mengetahui

Ketua Program Studi Infomatika/Ilmu Komputer

Issa Arwani, S.Kom, M.Sc

NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

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

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

Malang, 30 Juli 2016



Dewi Maida Safitri

NIM: 125150200111050

KATA PENGANTAR

Puji dan syukur senantiasa penulis haturkan kehadiran Allah SWT karena limpahan karunia dan rahmat-Nya penulis dapat menyelesaikan skripsi ini sebagai persyaratan dalam memperoleh gelar sarjana dengan judul “Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Genetika” dengan laçar. Dengan dibangunnya sistem ini diharapkan dapat membantu para peternak memberikan komposisi pakan dengan nutrisi yang tepat, murah, dan produksi yang tinggi.

Penulis menyadari bahwa dalam penyusunan skripsi ini tentunya tidak akan berjalan lancar tanpa adanya bantuan dari pihak lain, karena itu penulis ingin menyampaikan rasa terima kasih yang sangat besar kepada :

1. Imam Cholissodin, S.Si., M.Kom selaku dosen pembimbing 1 dan dosen pembimbing akademik yang telah memberikan ilmu dan masukan terkait pengerjaan skripsi ini.
2. Edy Santoso, S.Si., M.Kom selaku dosen pembimbing 2 yang telah memberikan ilmu dan masukan terkait pengerjaan skripsi ini.
3. Dr. Ir. Irfan H. Djunaidi, M.Sc selaku dosen Fakultas Peternakan yang berkenan menadi narasumber dan membantu terkait pengerjaan skripsi ini.
4. KUD Batu beserta peternak sapi perah di desa Dresel Batu yang telah menyediakan data guna pengerjaan skripsi ini.
5. Kedua orang tua penulis yang senantiasa memotivasi dan mendoakan terkait pengerjaan skripsi ini maupun dalam masa perkuliahan.
6. Teman-teman penulis Arinie Khaqqo, Az Zahra Rahma P.A, Fitri Bibi Suryani, Fina Af-Idatul H yang telah memberikan banyak dukungan berupa motivasi dan doa terkait pengerjaan skripsi ini.
7. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang tanpa bantuannya penulis tidak bisa menyelesaikan skripsi ini.

Penulis menyadari bahwa penlitian ini masih jauh dari kata sempurna. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun bagi penulis sehingga di lain waktu penulis dapat memberikan yang lebih baik lagi. Diharapkan dengan adanya penelitian ini dapat memberikan manfaat bagi semua civitas akademik.

Malang, 15 Mei 2016

Penulis

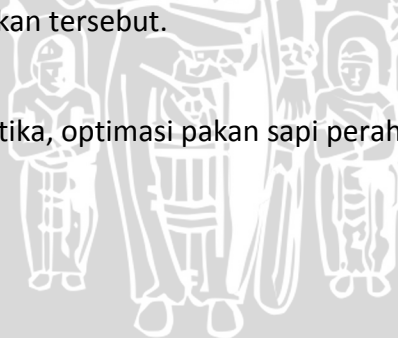
dewimayy@gmail.com

ABSTRAK

Pemberian komposisi pakan sapi yang tepat sangatlah penting dalam hal pemenuhan gizi sapi maupun dalam hal produksi. Namun harga bahan pakan sapi selalu menjadi kendala utama bagi peternak untuk memberikan komposisi yang tepat. Biaya pakan yang besar dan hasil produksi yang tidak sebanding merupakan permasalahan yang dialami sebagian besar peternak. Untuk meningkatkan pendapatan, biaya pakan harus dibuat sekecil mungkin namun bisa menghasilkan produksi susu yang sebanyak mungkin. Untuk mewujudkan tujuan tersebut digunakan algoritma genetika untuk membantu proses optimasi terhadap komposisi pakan sapi perah. Algoritma genetika telah terbukti dapat memberikan solusi yang mendekati optimal.

Representasi kromosom yang digunakan adalah *real code* dengan panjang kromosom sesuai dengan banyaknya jenis pakan, dimana setiap gen dalam kromosom bertindak sebagai bobot dari jenis pakan tersebut. Metode *crossover* yang digunakan adalah *extended intermediate crossover*, sedangkan metode mutasi yang digunakan adalah *random mutation* dan metode seleksi yang digunakan yakni *elitism selection*. Berdasarkan hasil pengujian terhadap parameter algoritma, didapatkan nilai *popsiz* 220, banyak generasi 1000, nilai *crossover rate* 0.6, nilai *mutation rate* 0.4, dan interval bobot gen adalah [1:80]. Parameter hasil pengujian tersebut akan digunakan untuk komputasi algoritma genetika. Hasil keluaran dari sistem adalah komposisi pakan dengan masing-masing bobotnya serta total harga dan estimasi pertambahan produksi yang dapat dihasilkan dari komposisi pakan tersebut.

Kata kunci : algoritma genetika, optimasi pakan sapi perah

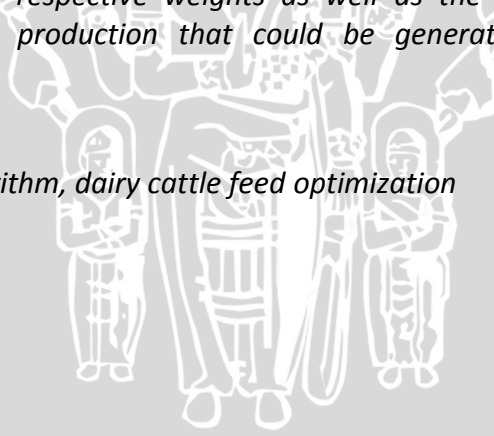


ABSTRACT

Giving proper dairy cattle feed composition is important to fulfill the requirement nutritions of dairy cattle or to maximize milk production. But the price of the feed ingredients has always been a major obstacle for farmer to provide the exact composition. High feed cost which not comparable with the production result is the problem experienced by most farmers. In order to increase the revenue, the cost of feed should be made as small as possible but still can produce milk as much as possible. To realize these goals, genetic algorithm is used to assist the optimization of dairy cows feed composition. Genetic algorithm have proven can provide near optimal solution.

The chromosome representation used in this research is real coded representation with the length of the chromosome is the number of types of feed, where every gene in chromosome acts as the weight of the feed types. The crossover method used is extended intermediate crossover, whereas the mutation method used is random mutation and the selection used is elitism selection. Based on the test results of the algorithm parameters, near optimal value of popsize is 220, the number of generation is 1000, crossover rate is 0.6, mutation rate is 0.4, and the weight interval is [1:80]. The parameter from the test result will be used for computing of the genetic algorithm. The output of the system is the feed composition with their respective weights as well as the total cost and the estimated incremental production that could be generated from the feed composition.

Keyword : *genetic algorithm, dairy cattle feed optimization*



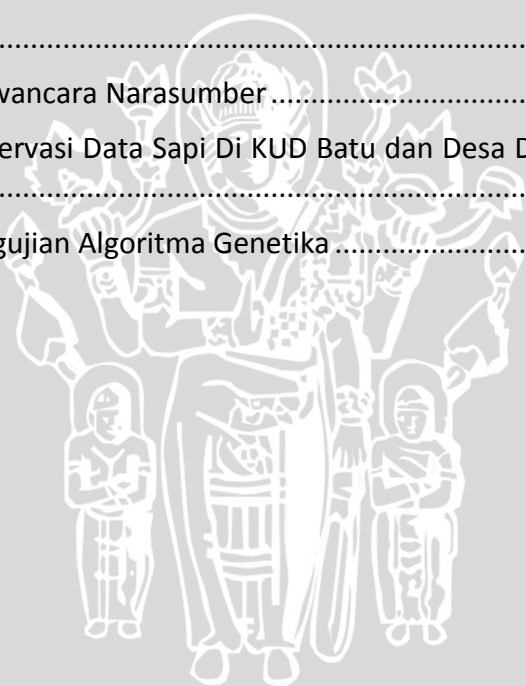
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Kebutuhan Nutrisi Sapi Perah.....	8
2.2.1 Kebutuhan Bahan Kering.....	9
2.2.2 Energi	10
2.3 Algoritma Genetika	12
2.3.1 Kromosom / Individu.....	13
2.3.2 <i>Fitness</i>	13
2.3.3 Populasi.....	13
2.4 Penerapan Algoritma Genetika	13
2.4.1 Parameter Algoritma Genetika	14
2.4.2 Representasi Kromosom	14
2.4.3 <i>Crossover</i>	15
2.4.4 Mutasi	16

2.4.5 Seleksi.....	17
BAB 3 METODOLOGI	18
3.1 Studi Literatur	19
3.2 Analisa Kebutuhan Sistem	19
3.3 Analisa dan Pengumpulan Data.....	19
3.4 Perancangan Sistem.....	20
3.5 Implementasi Sistem	20
3.6 Pengujian dan Evaluasi Sistem.....	21
3.7 Pengambilan Keputusan	21
BAB 4 PERANCANGAN.....	22
4.1 Formulasi Permasalahan.....	22
4.2 Alur Penyelesaian Menggunakan Algoritma Genetika	23
4.2.1 Membangun Individu dan Menentukan <i>Fitness</i>	24
4.2.2 Menentukan Populasi Awal	26
4.2.3 <i>Crossover</i>	27
4.2.4 Mutasi	30
4.2.5 Seleksi.....	31
4.3 Perancangan Pengujian Algoritma Genetika	33
4.3.1 Pengujian Interval Gen.....	33
4.3.2 Pengujian Ukuran Populasi	34
4.3.3 Pengujian Banyaknya Generasi.....	35
4.3.4 Pengujian Nilai <i>Crossover Rate</i> dan <i>Mutation Rate</i>	35
4.4 Perancangan Antarmuka	36
4.4.1 Rancangan Antarmuka Halaman Input Data	36
4.4.2 Rancanga Antarmuka Halaman Hasil Optimasi	37
BAB 5 IMPLEMENTASI	38
5.1 Implementasi Algoritma Genetika.....	38
5.1.1 Implementasi Penentuan Populasi Awal	38
5.1.2 Implementasi Proses <i>Crossover</i>	38
5.1.3 Implementasi Proses Mutasi.....	39
5.1.4 Implementasi Proses Seleksi	40
5.2 Implementasi Antarmuka Program	41



5.2.1 Implementasi Halaman Input Data	41
5.2.2 Implementasi Halaman Hasil Optimasi	42
BAB 6 PENGUJIAN DAN ANALISIS	44
6.1 Pengujian Interval Gen	44
6.2 Pengujian Ukuran Populasi	45
6.3 Pengujian Jumlah Generasi	47
6.4 Pengujian Nilai <i>Cr</i> dan <i>Mr</i>	48
6.5 Evaluasi Hasil Optimasi	50
BAB 7 PENUTUP	52
7.1 Kesimpulan	52
7.2 Saran	52
DAFTAR PUSTAKA	54
LAMPIRAN A Hasil Wawancara Narasumber	56
LAMPIRAN B Hasil Observasi Data Sapi Di KUD Batu dan Desa Dresel, Kecamatan Batu, Kota Batu	58
LAMPIRAN C Hasil Pengujian Algoritma Genetika	61



DAFTAR TABEL

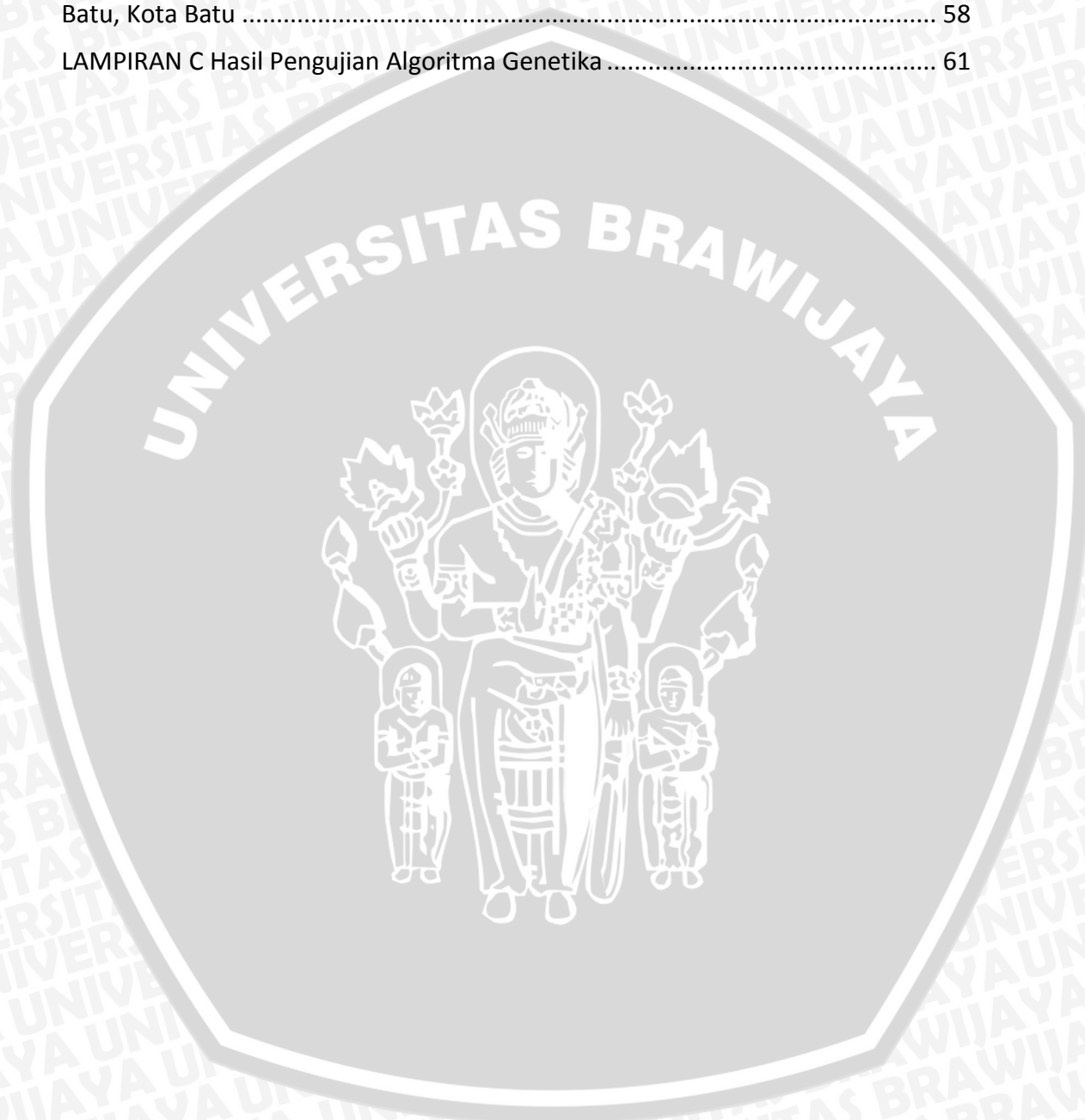
Tabel 2.1 Kajian Pustaka	6
Tabel 2.2 Tabel Panduan Perhitungan Bahan Kering Maksimal	9
Tabel 2.3 Tabel Kebutuhan Nutrisi Sapi Perah Laktasi.....	10
Tabel 2.4 Tabel Kebutuhan Nutrisi per Liter Produksi Susu.....	11
Tabel 2.5 Tabel Perbandingan Kebutuhan Berdasarkan Produksi.....	12
Tabel 4.1 Tabel Kandungan Nutrisi Bahan Pakan Sapi Perah	22
Tabel 4.2 Tabel Perbandingan Nutrisi Sapi dan Pakan	26
Tabel 4.3 Inisialisasi Populasi Awal	27
Tabel 4.4 Kromosom Terpilih Untuk <i>Crossover</i>	29
Tabel 4.5 Kromosom Hasil <i>Crossover</i>	29
Tabel 4.6 Kromosom Terpilih Untuk <i>Crossover</i>	29
Tabel 4.7 Kromosom Hasil <i>Crossover</i>	30
Tabel 4.8 Kromosom terpilih untuk proses mutasi.....	31
Tabel 4.9 Kromosom hasil proses mutasi	31
Tabel 4.10 Individu total dalam sebuah populasi	32
Tabel 4.11 Individu dengan <i>fitness</i> tertinggi.....	33
Tabel 4.12 Rancangan Pengujian Interval Gen	34
Tabel 4.13 Rancangan Pengujian Ukuran Populasi.....	34
Tabel 4.14 Rancangan Pengujian Jumlah Generasi.....	35
Tabel 4.15 Rancangan Pengujian Nilai Cr dan Mr.....	36
Tabel 6.1 Hasil Pengujian Interval Gen	44
Tabel 6.2 Hasil Pengujian Ukuran Populasi.....	46
Tabel 6.3 Hasil Pengujian Banyaknya Generasi	47
Tabel 6.4 Hasil Pengujian Kombinasi Nilai Cr dan Mr	48
Tabel 6.5 Data Hasil Observasi Peternak	50
Tabel 6.6 Hasil Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Genetika	51
Tabel 6.7 Perbandingan Komposisi Peternak dan Sistem.....	51

DAFTAR GAMBAR

Gambar 2.1 Peternakan Sapi Perah	8
Gambar 2.2 Bekatul dan polar pakan konsenstrat sapi perah	9
Gambar 2.3 <i>General Pseudococe</i> Algoritma Genetika	13
Gambar 2.4 Ilustrasi <i>Single-point Crossover</i>	15
Gambar 2.5 Ilustrasi <i>Two-point Crossover</i>	16
Gambar 2.6 Ilustrasi <i>Interchanging Mutation</i>	16
Gambar 3.1 Diagram Alir Metodologi Penelitian	18
Gambar 3.2 Alur Kerja Sistem	20
Gambar 4.1 Flowchart Penyelesaian Algoritma Genetika	23
Gambar 4.2 Contoh Representasi Kromosom	24
Gambar 4.3 Flowchart Proses Pembentukan Populasi Awal	27
Gambar 4.4 Flowchart Proses <i>Crossover</i>	28
Gambar 4.5 Flowchart Proses Mutasi	30
Gambar 4.6 Flowchart Proses Seleksi	32
Gambar 4.7 Rancangan Antarmuka Halaman Input Data	37
Gambar 4.8 Rancangan Antarmuka Halaman Hasil Perhitungan	37
Gambar 5.1 <i>Source Code</i> Penentuan Populasi Awal	38
Gambar 5.2 <i>Source Code</i> Proses <i>Crossover</i>	39
Gambar 5.3 <i>Source Code</i> Proses Mutasi	40
Gambar 5.4 <i>Source Code</i> Proses Seleksi	40
Gambar 5.5 Implementasi Halaman Input Data	41
Gambar 5.6 Halaman Pilih Pakan	42
Gambar 5.7 Implementasi Halaman Hasil Optimasi	43
Gambar 6.1 Grafik Hasil Pengujian Interval Gen	45
Gambar 6.2 Grafik Hasil Pengujian Ukuran Populasi	46
Gambar 6.3 Grafik Pengujian Jumlah Generasi	48
Gambar 6.4 Grafik Hasil Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i>	49

DAFTAR LAMPIRAN

LAMPIRAN A Hasil Wawancara Narasumber	56
LAMPIRAN B Hasil Observasi Data Sapi Di KUD Batu dan Desa Dresel, Kecamatan Batu, Kota Batu	58
LAMPIRAN C Hasil Pengujian Algoritma Genetika	61



BAB 1 PENDAHULUAN

1.1 Latar belakang

Susu menjadi komoditas paling penting bagi industri pangan terutama dalam pemenuhan gizi masyarakat. Banyaknya kandungan nutrisi dan manfaat yang bisa didapat dari susu meningkatkan kesadaran masyarakat akan pentingnya mengkonsumsi susu. Hal ini mengakibatkan meningkatnya angka permintaan susu di Indonesia. Namun meningkatnya permintaan susu ini tidak diimbangi dengan produksi yang ada. Dalam periode tahun 2002 hingga 2007 angka permintaan susu naik mencapai 14,01% sedangkan sektor produksi susu hanya mengalami peningkatan tidak lebih dari 2% di periode yang sama. Impor susu yang dilakukan Indonesia semakin mendukung bahwa produksi susu di Indonesia belum bisa memenuhi kebutuhan susu nasional (Farid & Sukesi, 2011).

Data Badan Pusat Statistika mengenai produksi susu di Indonesia pada tahun 2009 hingga 2011 mengalami fluktuasi dengan produksi susu tertinggi pada tahun 2011 mencapai 974 ribu ton dan produksi terus menurun hingga tahun 2015. Di Jawa Timur sendiri produksi susu sapi perah mengalami penurunan yang cukup drastis dari produksi susu sekitar 55 ribu ton pada tahun 2012 menurun hingga 41 ribu ton pada tahun 2013 dan masih belum menunjukkan kenaikan hingga sekarang. Dan angka produksi tersebut belum bisa memenuhi baik kebutuhan untuk industri maupun konsumsi penduduk. Hal tersebut membuktikan perlu adanya upaya untuk meningkatkan produksi susu sapi perah (Statistik, 2015). Meningkatkan produksi susu yang dihasilkan melalui sapi perah merupakan salah satu tantangan yang dihadapi Industri Pengolahan Susu. Hal ini juga didorong oleh Peraturan Presiden nomor 28 tahun 2008 tentang kebijakan industri pengolahan susu dimana salah satu kebijakannya yakni meningkatkan produktivitas ternak sapi yang rata-rata menghasilkan 8-12 liter per ekor/hari menjadi 20 liter per ekor/hari (Yudhoyono, 2008).

Permasalahan utama yang dihadapi peternak dalam memproduksi susu yakni ketersediaan pakan. Terbatasnya lahan menjadikan peternak semakin susah untuk memproduksi sendiri hijauan bagi ternaknya. Ditambah lagi dengan naiknya harga bahan baku pakan yang berdampak pada naiknya harga konsentrat (Mansyur, et al., 2008). Untung ruginya yang dialami peternak sering kali dipengaruhi oleh cara pemberian pakan yang digunakan. Tak jarang para peternak memberikan pakan mahal dengan tujuan menambah nutrisi namun ternyata tidak menambah produksi (Miller, 1979). Untuk itu peternak harus pintar-pintar meramu komposisi pakan yang tepat. Dimana dengan biaya pakan yang sekecil mungkin harus bisa menghasilkan produk sebanyak mungkin.

Penentuan komposisi pakan yang tepat adalah kunci dalam pemenuhan nutrisi pada sapi perah guna mendapatkan produksi yang maksimal. Penelitian dalam hal optimasi komposisi telah banyak dilakukan dan berbagai macam metode telah diterapkan guna mendapatkan komposisi yang tepat. Zhao, dkk (2015) menerapkan *Merging Genetic Algorithm* dan *Ant Colony Algorithm* dalam

penelitiannya untuk menentukan komposisi web service guna membangun web service yang efisien (Zhao, et al., 2015). Penelitian optimasi komposisi pakan juga dilakukan oleh Muliantara (2012) dalam mengoptimasi komposisi bahan pakan ikan lele. Peneliti memandang permasalahan sebagai *multi objective problem* karena penelitian tersebut menggabungkan dua tujuan yang berbeda yakni memperoleh kandungan gizi yang baik dan mendapatkan harga yang efisien. Dalam permasalahan *multi object* peneliti menggunakan *Multi Objective Optimization with Invasive Weed Optimization-Subtractive Clustering* untuk mendapatkan komposisi bahan pakan yang tepat (Muliantara, 2012).

Penelitian tentang optimasi pakan sapi perah sudah pernah dilakukan sebelumnya oleh Fakhroh (2015). Dalam penelitiannya Fakhroh juga menggunakan algoritma genetika untuk menentukan komposisi pakan yang tepat. Tujuan optimasi pakan yang dilakukan peneliti adalah untuk mendapatkan komposisi pakan yang tepat dalam pemenuhan gizi sapi dengan harga yang sekecil mungkin (Fakhroh, et al., 2015). Berdasarkan penelitian yang dilakukan Fakhroh (2015), akan dibangun sistem untuk mengoptimasi pakan sapi perah dengan menekankan pada aspek produksi susu. Penelitian ini tidak hanya akan menentukan komposisi pakan untuk pemenuhan gizi sapi saja, namun juga menitik beratkan pada nutrisi yang dibutuhkan untuk meningkatkan produksi susu dengan biaya pakan seminimal mungkin. Penelitian ini akan dibangun menggunakan algoritma genetika karena dalam penelitian-penelitian sebelumnya algoritma genetika mampu memberikan solusi yang hampir mendekati optimal dalam hal optimasi komposisi dan dapat menjangkau range pencarian yang cukup besar.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang didapat adalah sebagai berikut :

1. Bagaimana mengimplementasikan algoritma genetika dalam mengoptimasi komposisi pakan yang tepat untuk sapi perah?
2. Bagaimana menentukan parameter algoritma genetika yang tepat untuk mendapatkan komposisi pakan sapi perah yang baik?
3. Bagaimana tingkat kualitas solusi yang dihasilkan algoritma genetika?

1.3 Tujuan

Tujuan yang ingin dicapai dengan dilakukannya penelitian ini adalah sebagai berikut :

1. Mengimplementasikan algoritma genetika untuk mengoptimasi komposisi pakan yang tepat bagi sapi perah.
2. Menentukan parameter algoritma genetika yang tepat untuk mendapatkan komposisi pakan sapi perah yang baik.
3. Mengetahui tingkat kualitas solusi yang dihasilkan algoritma genetika.

1.4 Manfaat

Harapan dilakukannya penelitian ini adalah tentunya dapat memberikan manfaat bagi beberapa pihak terkait. Manfaat yang bisa didapatkan dengan adanya penelitian ini adalah :

- Bagi peternak sapi perah

Penelitian ini dapat memeberikan solusi bagi para peternak sapi perah dalam meningkatkan produksi susu. Dengan penelitian ini para peternak dapat meningkatkan produksi susu tanpa harus mengeluarkan biaya yang tinggi untuk komsumsi pakan sapi perah.

- Bagi pemerintah

Penelitian ini dapat membatu pemerintah untuk meningkatkan angka produksi susu nasional.

1.5 Batasan masalah

Batasan permasalahan yang akan diterapkan dalam penelitian ini adalah :

1. Data sapi dan pakan yang digunakan pada penelitian ini adalah data yang diperoleh dari peternak sapi perah di Kota Batu dan KUD Batu.
2. Penentuan komposisi pakan sapi didasarkan pada berat badan sapi, kandungan lemak sapi, dan rata-rata produksi susu per ekor per hari.
3. Dalam penelitian ini terdapat 10 jenis pakan sapi perah yang akan dioptimasi.

1.6 Sistematika pembahasan

Sistematika penulisan dalam penelitian ini dapat dirumuskan sebagai berikut :

BAB I Pendahuluan

Pada bab ini dijelaskan permasalahan yang melatar belakangi dilakukannya penelitian ini, rumusan permasalahan dalam penelitian, tujuan dilakukannya penelitian, manfaat yang bisa didapat dengan adanya penelitian, dan sistematika penulisan dari penelitian yang akan dilakukan.

BAB II Landasan Kepustakaan

Pada bab ini dijelaskan teori-teori mengenai objek-objek dalam penelitian yang akan menjadi acuan dilakukannya penelitian. Kajian pustaka dan teori yang dijelaskan didapatkan dari literatur, website pemerintahan, dan jurnal-jurnal nasional maupun internasional.

BAB III Metodologi

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam membangun penelitian ini. Langkah-langkah yang dilakukan yakni meliputi studi literatur, analisa dan pengumpulan data, analisa dan perancangan sistem, implementasi, pengujian, dan pengambilan keputusan.

BAB IV Perancangan

Pada bab ini dijelaskan rancangan sistem yang akan dibangun. Perancangan yang akan dilakukan meliputi perancangan antarmuka, perancangan pengujian.

BAB V Implementasi

Pada bab ini dijelaskan teknik implementasi yang digunakan. Tahapan dari proses implementasi akan terdiri dari spesifikasi sistem, batasan implementasi, implementasi algoritma, dan implementasi antarmuka.

BAB VI Pengujian dan Analisis

Pada bab ini dijelaskan pengujian yang dilakukan terhadap sisem yang sudah dibangun. Pengujian yang akan dilakukan meliputi pengujian populasi, pengujian generasi, dan pengujian nilai cr dan mr . Serta akan dijelaskan pengambilan keputusan berdasarkan pengujian yang telah dijalankan

BAB VIII Penutup

Pada bab ini dijelaskan kesimpulan dari penelitian yang telah dijalankan. Kesimpulan pada bab ini akan menjawab rumusan permasalahan yang telah disusun sebelumnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan semua kajian pustaka dan teori mengenai objek-objek yang akan digunakan dalam penelitian ini. Dasar teori dalam bab ini akan membahas tentang nutrisi kebutuhan sapi perah dan penyusunan ransum sapi perah dan juga tentang algoritma genetika sebagai metode yang akan digunakan dalam penyelesaian permasalahan.

2.1 Kajian Pustaka

Pada sub bab kajian pustaka ini akan dijelaskan beberapa penelitian yang terkait dengan penelitian yang akan dilakukan. Penelitian yang akan dijelaskan meliputi penelitian terkait algoritma genetika sebagai metode dalam penyelesaian permasalahan optimasi komposisi.

Penelitian terkait objek komposisi pakan sapi perah sebelumnya pernah dilakukan oleh Fakhroh (2015). Dalam penelitian ini Fakhroh menerapkan algoritma genetika untuk mendapatkan komposisi pakan sapi. Penelitian tersebut dilakukan untuk meningkatkan pendapatan peternak dengan cara menekan biaya pakan namun tetap menjaga kebutuhan nutrisi sapi. Dalam penelitiannya, Fakhroh menggunakan representasi *real code* untuk mewakili bobot setiap jenis bahan pakan dengan *extended intermediate* sebagai metode *crossover* dan *random mutation* sebagai metode mutasi. Proses seleksi yang digunakan menggunakan metode *elitism* sehingga individu yang terseleksi adalah individu yang memiliki nilai *fitness* tertinggi pada populasinya (Fakhroh, et al., 2015).

Penelitian sejenis juga telah dilakukan namun pada objek yang berbeda. Penelitian yang dilakukan oleh Wahid (2015) yakni untuk menentukan komposisi makanan bagi penderita kolesterol. Metode yang digunakan adalah algoritma genetika dengan menerapkan *single point crossover* sebagai metode *crossover* dan proses mutasi menggunakan *reciprocal exchange mutation*. Penelitian tersebut bertujuan untuk mendapatkan komposisi makanan dengan kandungan lemak yang tepat dengan biaya yang seminimal mungkin. Representasi yang digunakan dalam algoritma genetika tersebut yakni representasi permutasi dimana representasi tersebut akan mewakili urutan bahan makanan. Hasil yang didapatkan dari penelitian tersebut bisa dikatakan adalah hasil yang paling baik karena proses seleksi yang digunakan adalah seleksi *elitism* sehingga semua individu yang terpilih untuk generasi selanjutnya merupakan individu terbaik dari generasi sebelumnya (Wahid & Mahmudy, 2015).

Penelitian lainnya juga dilakukan oleh Marginingtyas (2015) dalam mengoptimasi komposisi pakan ayam petelur. Penelitian tersebut menggunakan algoritma genetika dalam mencari komposisi pakan yang tepat. Dalam penelitiannya Marginingtyas menggunakan representasi *real code* dengan *gen* *random* pada interval 1-10 yang akan mewakili bobot tiap bahan pakan. Metode *crossover* yang digunakan yakni *one cut point* sedangkan metode mutasi yang digunakan yakni *reciprocal exchange*. Tujuan dari penelitian ini yakni

mendapatkan komposisi pakan dengan nilai gizi sesuai dengan kebutuhan ayam dengan meminimalkan biaya pakan. Dari hasil penelitian tersebut didapatkan komposisi pakan yang optimal karena seleksi yang digunakan adalah seleksi *elitism* dimana individu yang dipilih untuk generasi selanjutnya merupakan generasi dengan nilai fitness paling tinggi (Marginingtyas, et al., 2015).

Penelitian berikutnya dilakukan oleh Li, dkk (2015) dalam menemukan pemilihan komposisi *suplier auto part* yang optimal. Penelitian ini menggunakan metode adaptive algoritma genetika dan menggunakan representasi biner dalam pemecahan permasalahan mereka. Proses *crossover* dan mutasi yang mereka gunakan adalah dengan menyesuaikan proses *crossover* dan mutasinya secara dinamis dengan menghitung probabilitas pada kedua proses. Penelitian ini menggunakan metode *roulette wheel* untuk metode seleksinya dengan memilih acak individu dari setiap populasi. Tujuan dari penelitian ini adalah untuk menemukan komposisi *suplier auto part* yang optimal berdasarkan variabel yang telah ditentukan (Li, et al., 2015).

Penelitian terakhir dilakukan oleh Wang, dkk (2013) dalam menemukan komposisi layanan data-intensif efektif yakni dengan mengurangi biaya penyusunan layanan. Penelitian ini menggunakan metode *improved* algoritma genetika untuk penyelesaian masalahnya. Representasi kromosom yang digunakan dalam metode ini adalah skema *array integer* dengan metode *crossover* yang digunakan adalah metode *single point crossover*. Namun metode yang mutasi yang digunakan telah dimodifikasi sendiri oleh peneliti. Proses seleksi yang dilakukan dalam penelitian ini menggunakan gabungan dari metode *elitism* dan *tournament*. Tujuan dari penelitian ini yakni mengurangi biaya penyusunan *workflow* data-intensif dengan mengoptimalkan biaya transfer data, penempatan data, dan penyimpanan data (Wang, et al., 2013).

Tabel 2.1 Kajian Pustaka

No.	Judul	Objek	Metode	Hasil
1.	Penelitian mengenai optimasi terhadap komposisi pakan sapi perah.	Kebutuhan nutrisi sapi perah dan komposisi pakan	Algoritma Genetika	Permasalahan penelitian ini adalah naiknya harga pakan ternak sapi perah yang tidak sebangun dengan pemberian pakan terhadap sapi perah yang tidak efisien. Hasil penelitian ini menunjukkan bahwa algoritma genetika mampu memberikan komposisi pakan yang optimal dengan harga serendah mungkin namun tetap memenuhi kebutuhan gizi sapi perah.
2.	Penelitian mengenai optimasi	Asupan energi dan	Algoritma Genetika	Permasalahan dalam penelitian ini adalah sulitnya

	terhadap komposisi makanan bagi penderita kolesterol.	komposisi bahan makanan		menentukan komposisi makan pagi, makan siang, dan makan malam bagi penderita kolesterol. Hasil penelitian ini menunjukkan bahwa algoritma genetika mampu memberikan komposisi optimal sesuai dengan panduan gizi bagi penderita kolesterol dengan menu lengkap.
3.	Penelitian mengenai optimasi terhadap komposisi pakan untuk ayam petelur.	Komposisi pakan ayam	Algoritma Genetika	Permasalahan dalam penelitian ini adalah besarnya biaya yang harus dikeluarkan untuk memenuhi kebutuhan pakan ayam petelur. Sehingga peternak membuat komposisi yang tidak baik hanya untuk meminimalkan biaya pakan. Hasil penelitian ini menunjukkan algoritma genetika mampu menghasilkan komposisi pakan untuk ayam petelur dengan biaya minimum tanpa mengurangi enam nutrisi yang dibutuhkan oleh ayam petelur.
4.	<i>Study on Auto Parts Suppliers Composition Selection Based on Adaptive Genetic Algorithm</i>	Daftar supplier penyedia auto parts	Adaptif Algoritma Genetika	Permasalahan dalam penelitian ini adalah untuk menentukan pemilihan komposisi <i>supplier auto parts</i> . Hasil dari penelitian ini menunjukkan bahwa algoritma genetika dapat menemukan solusi komposisi <i>supplier</i> yang hampir mendekati optimal. Kemudian menggunakan adaptive algoritma genetika dengan menyesuaikan operator <i>crossover</i> dan mutasinya. Hasilnya adaptive algoritma genetika mempercepat tingkat konvergensi dan juga mempercepat eksekusi algoritma.

5.	<i>An Improved Genetic Algorithm for Cost-Effective Data Intensive Service Composition</i>	Layanan data intensif	Improved Genetic Algorithm	Penelitian ini berfokus pada pengurangan biaya yang digunakan untuk menyusun alur kerja sebuah layanan data-intensif. Hasil dari penelitian ini menunjukkan bahwa seleksi <i>elitism</i> dan <i>tournament</i> dan <i>modified crossover</i> dan mutasi yang digunakan dapat mengurangi biaya komposisi yang meliputi transfer data, penempatan data, dan penyimpanan data.
----	--	-----------------------	----------------------------	---

Sumber : Diadaptasi dari (Fakhroh, et al., 2015), (Wahid & Mahmudy, 2015), (Marginingtyas, et al., 2015), (Li, et al., 2015), (Wang, et al., 2013)

2.2 Kebutuhan Nutrisi Sapi Perah

Menentukan kebutuhan nutrisi sapi perah memerlukan dua langkah besar dengan beberapa modifikasi pada setiap nutrisi yang berbeda. Pertama yakni membangun jumlah dari nutrisi yang digunakan sapi untuk setiap aktivitas tertentu. Kedua yakni menghitung nutrisi tersebut. Dalam menghitung kebutuhan nutrisi tersebut, peternak harus memperhatikan efisiensi setiap nutrisi yang akan digunakan untuk setiap aktivitas yang berbeda. Jumlah nutrisi yang dibutuhkan dibangun berdasarkan jenis aktivitasnya yang meliputi pemeliharaan, pertumbuhan/penggemukan, memproduksi susu, reproduksi, dan bekerja (seperti berjalan, dan lain-lain). Setiap ekor sapi memiliki kebutuhan pemeliharaan, sedangkan aktivitas lainnya bergantung pada situasi sapi itu sendiri. Sebagian besar pakan yang diberikan pada ternak sapi akan diubah menjadi energi. Setiap kegiatan, proses, maupun reaksi biokimia pada tubuh sapi membutuhkan energi terlepas kondisi sapi sedang dalam kondisi laktasi maupun kering (Miller, 1979).



Gambar 2.1 Peternakan Sapi Perah

2.2.1 Kebutuhan Bahan Kering

Konsumsi bahan kering merupakan komponen penting dalam pemenuhan nutrisi sapi karena bahan kering merupakan penyedia nutrisi yang digunakan dalam menjaga kesehatan dan melakukan produksi. Komposisi bahan kering yang akurat sangatlah penting guna mencegah terjadinya kelebihan dan kekurangan nutrisi pada sapi. Kekurangan nutrisi pada sapi akan menyebabkan terganggunya kesehatan sapi dan jalannya produksi susu, dan kelebihan pemberian nutrisi akan berdampak pada pembengkakan biaya pakan. Penentuan konsumsi bahan kering pada setiap ekor sapi pada masa laktasi berbeda. Umumnya masa laktasi sapi perah berkisar dari 1 hingga 40 minggu (Council, 2001).



Gambar 2.2 Bekatul dan polar pakan konsenstrat sapi perah

Konsumsi bahan kering setiap ekor sapi dapat dihitung dengan menggunakan nilai 4% FCM kemudian membandingkan nilai tersebut dengan tabel panduan bahan kering (Council, 1978). Formula Gaines mengenai 4% FCM (*Fat Corrected Milk*) telah digunakan lebih dari 70 tahun. Formula ini digunakan untuk menyamakan hasil susu dalam energi dasar. Formula Gaines didasarkan pada asumsi bahwa konsentrasi NE_L pada susu yang mengandung 4% lemak adalah 0.749 Mcal/kg dan energi pada susu dengan lemak dibawah 3% tidak akan dihitung. Perhitungan 4% FCM dapat dihitung dengan persamaan 2.1 (Council, 2001):

$$4\% \text{ FCM} = 0.4 \times \text{milk}(\text{kg}/\text{day}) + 15(\text{fat} \times \text{milk}) \quad (2.1)$$

Dari hasil perhitungan 4% FCM tersebut dicari nilai persentase maksimum berdasarkan berat badan sapi dengan menggunakan Tabel 2.2.

Tabel 2.2 Tabel Panduan Perhitungan Bahan Kering Maksimal

4% FCM	Berat Badan (kg)				
	400	500	600	700	800
10	2.5	2.3	2.2	2.1	2
15	2.8	2.5	2.4	2.3	2.2
20	3.1	2.8	2.7	2.6	2.4

4% FCM	Berat Badan (kg)				
	400	500	600	700	800
25	3.4	3.1	3	2.8	2.6
30	3.7	3.4	3.2	3	2.8
35	4	3.5	3.4	3.2	3
40	-	3.8	3.6	3.4	3.2
45	-	4	3.8	3.6	3.4

Sumber : (Council, 1978)

Hasil bilangan dari Tabel 2.2 merupakan persentase dari bobot sapi yang hasil akhirnya merupakan nilai bahan kering yang dapat dikonsumsi oleh sapi tersebut.

2.2.2 Energi

Energi yang biasanya dibutuhkan seekor sapi yakni meliputi energi untuk pertumbuhan dan energi untuk produksi. Menurut NRC (1978) satuan energi yang paling sering digunakan adalah Total Digestible Nutrient (TDN). Penggunaan TDN sering digunakan karena sebegini besar data yang tersedia baik data kebutuhan nutrisi sapi perah maupun data ketersediaan nutrisi pakan adalah data TDN (Council, 1978). Nutrisi yang tidak kalah pentingnya yakni protein dalam menyediakan asam amino bagi sapi yang sangat dibutuhkan untuk beberapa proses sistetik dalam tubuh. Protein sangat dibutuhkan oleh sapi untuk pemeliharaan, pertumbuhan, reproduksi, dan laktasi. Protein yang digunakan dalam perhitungan kebutuhan nutrisi adalah *digestible protein* dan direpresentasikan dengan total protein kasar. Protein kasar digunakan karena dianggap paling akurat jika dikonversi dari tabel komposisi ke perhitungan kebutuhan diet sapi (Council, 1978).

NRC (1978) telah mengembangkan beberapa sistem net energy yang terdiri dari Net Energy of Maintenance (NE_M) yang digunakan untuk menghitung kebutuhan energi sapi non laktasi, Net Energy of Gain (NE_G) yang digunakan untuk mengukur energi pada sapi non laktasi yang sedang dalam masa pertumbuhan, dan Net Energy of Lactation (NE_L) yang digunakan untuk mengukur energi pada sapi laktasi. Karena energi yang digunakan untuk pemeliharaan dan produksi pada sapi perah hampir sama, maka kebutuhan energi pada sapi perah dapat dihitung menggunakan satu sistem net energy saja yakni NE_L baik untuk energi pemeliharaan maupun produksi (Council, 1978).

Untuk mengukur kebutuhan energi pada sapi laktasi dapat dilihat pada tabel kebutuhan yang dikembangkan oleh NRC (1978). Kebutuhan nutrisi setiap sapi perah berbeda dimana kebutuhan hidup pokok dibedakan berdasarkan berat badan setiap ekor sapi perah. Untuk mencari kebutuhan hidup pokok sapi perah dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel Kebutuhan Nutrisi Sapi Perah Laktasi

BB (Kg)	NE_L (Mcal/kg)	TDN (Kg)	PK (kg)
350	6.47	2.85	0.341

BB (Kg)	NE _L (Mcal/kg)	TDN (Kg)	PK (kg)
400	7.16	3.15	0.373
450	7.82	3.44	0.403
500	8.46	3.72	0.432
550	9.09	4.00	0.461
600	9.70	4.27	0.489

Sumber : (Council, 1978)

Untuk menghitung kebutuhan nutrisi produksi sapi perah dibedakan berdasarkan kandungan lemak dalam produksi susu. Tabel 2.4 merupakan kandungan nutrisi yang dibutuhkan untuk memproduksi satu liter susu dengan lemak tertentu. Untuk mengetahui kebutuhan nutrisi produksi dapat dilihat pada Tabel 2.4.

Tabel 2.4 Tabel Kebutuhan Nutrisi per Liter Produksi Susu

Kadar Lemak (%)	NE _L (Mcal)	TDN (Kg)	PK (kg)
2.5	0.59	0.260	0.072
3	0.64	0.282	0.077
3.5	0.69	0.304	0.082
4	0.74	0.326	0.087
4.5	0.78	0.344	0.092
5	0.83	0.365	0.098
5.5	0.88	0.387	0.103
6	0.93	0.41	0.108

Sumber : (Council, 1978)

Kebutuhan nutrisi untuk produksi tidak hanya ditentukan oleh ketiga nutrisi seperti pada Tabel 2.4, namun juga ditentukan oleh nutrisi BK. Perhitungan BK untuk produksi ini berlaku untuk umum atau tidak memperhatikan kandungan lemak pada susu sapi. Menurut penelitian yang dilakukan oleh Novianti (2014) didapatkan bahwa 0.2 – 0.9 kg BK/ekor/hari dapat meningkatkan produksi susu sebanyak 0.2 – 0.5 liter/ekor/hari. Dari kesimpulan tersebut bahwa untuk menghasilkan satu liter susu, seekor sapi setidaknya membutuhkan 1.57 kg BK (Novianti, 2014).

NRC (1978) juga mengembangkan tabel untuk rasio perbandingan kebutuhan nutrisi sapi perah berdasarkan jumlah produksi susu. Selain dengan menggunakan rumus 4% FCM perhitungan kebutuhan BK sapi perah juga dapat dihitung dengan membagi total kebutuhan NE_L sapi perah dengan nilai rasio pada Tabel 2.5. Perbandingan rasio tersebut dapat dilihat pada Tabel 2.5.

Tabel 2.5 Tabel Perbandingan Kebutuhan Berdasarkan Produksi

Produksi Susu (L)	NE _L / kg BK (Mkal/kg)
< 8	1.42
8 – 13	1.52
13 – 18	1.62
> 18	1.72

Sumber : (Council, 1978)

2.3 Algoritma Genetika

Algoritma genetika pertama kali dikenalkan oleh John H. Holland pada tahun 1975. Algoritma genetika menjadi metode yang cukup populer dalam evolusi komputasi yang seringnya digunakan untuk menyelesaikan masalah dalam hal pencarian hasil optimal atau mendekati optimal dengan rentang pencarian yang cukup besar. Algoritma genetika menerapkan konsep dari prinsip biologi dalam mensimulasikan proses evolusi di lingkungan atau dengan kata lain seleksi alam (Fanjiang & Syu, 2014). Algoritma genetika memiliki beberapa fitur penting. Pertama algoritma genetika merupakan algoritma stokastik yang berarti elemen utama dalam algoritma tersebut adalah pengacakan. Baik proses reproduksi dan seleksi dalam algoritma genetika memerlukan prosedur pengacakan. Kedua algoritma genetika selalu memberikan sekumpulan solusi dalam sebuah populasi. Algoritma genetika dapat mengkombinasikan beberapa solusi yang berbeda untuk mendapatkan solusi yang lebih baik (Sivanandam & Deepa, 2008).

Proses evolusi pada algoritma evolusi, solusi yang menjadi kandidat dalam permasalahan yang akan diselesaikan disebut sebagai kromosom atau individu. Sebuah kromosom mengandung gen-gen yang dibentuk dalam beberapa format. Dalam setiap generasi dalam algoritma genetika memiliki sejumlah kromosom, dan seluruh kromosom dalam suatu generasi disebut dengan populasi. Jumlah kromosom dalam suatu generasi dan banyaknya generasi dalam algoritma genetika ditentukan oleh pengguna algoritma genetika itu sendiri pada saat proses perancangan. Elemen lain yang cukup penting dalam algoritma genetika yakni fitness yang digunakan untuk mengevaluasi kromosom pada permasalahan yang akan diselesaikan (Fanjiang & Syu, 2014).

Beberapa individu dalam algoritma genetika akan menjalani transformasi atau operasi genetik guna mendapatkan individu baru. Dalam algoritma genetika terdapat dua jenis transformasi yakni mutasi dimana proses ini akan menghasilkan individu baru dengan mengubah satu individu, dan *crossover* dimana proses ini akan menghasilkan individu baru dengan mengkombinasikan sebagian dari dua individu yang berbeda. Individu baru hasil proses reproduksi tersebut dengan *offspring* yang nantinya juga akan ikut dalam proses evaluasi. Untuk mendapatkan populasi baru, algoritma genetika melakukan proses seleksi pada populasi *parent* dan populasi *offspring* berdasarkan fitnessnya. Setelah beberapa generasi,

algoritma genetika akan sampai pada individu terbaiknya yang mana diharapkan akan merepresentasikan solusi yang optimal atau mendekati optimal (Gen & Cheng, 2000). Struktur dari algoritma genetika dapat dilihat pada Gambar 2.3.

```
Procedure: Genetic Algorithm
begin
  t ← 0;
  initialize P(t);
  evaluate P(t);
  while (not termination condition) do
  begin
    recombine P(t) to yield C(t);
    evaluate C(t);
    select P(t+1) from P(t) and C(t);
    t ← t+1;
  end
end
```

Gambar 2.3 General Pseudocode Algoritma Genetika

Sumber : (Gen & Cheng, 2000)

2.3.1 Kromosom / Individu

Sebuah kromosom merepresentasikan sebuah solusi. Sebuah kromosom terdiri dari beberapa gen dimana gen yang akan mendeskripsikan kemungkinan solusi pada sebuah permasalahan tanpa harus menjadi solusi itu sendiri. Sebuah kromosom harus mengandung informasi tentang solusi yang direpresentasikan. Setiap kromosom yang dibentuk harus bersifat *unique* satu dengan yang lain (Sivanandam & Deepa, 2008).

2.3.2 Fitness

Fitness dari setiap individu pada algoritma genetika mewakili nilai dari kromosom tersebut. Nilai *fitness* tidak hanya mengindikasikan seberapa baik solusi tersebut, tetapi juga berperan dalam menentukan seberapa dekat kromosom tersebut dengan kromosom optimal yang akan menjadi solusi terakhir nantinya (Sivanandam & Deepa, 2008).

2.3.3 Populasi

Populasi merupakan sekumpulan dari beberapa individu. Sebuah populasi mengandung beberapa individu/kromosom yang selanjutnya akan diuji. Dalam algoritma genetika terdapat dua aspek penting yang dimiliki oleh sebuah populasi. Yang pertama yakni populasi awal yang akan mengawali sebuah generasi. Dan yang kedua ukuran populasi itu sendiri. Populasi awal pada generasi awal biasanya dibangkitkan secara acak (Sivanandam & Deepa, 2008).

2.4 Penerapan Algoritma Genetika

Dalam penerapan algoritma genetika, peneliti harus memperhatikan beberapa hal terkait di dalamnya. Sebelum melakukan penelitian peneliti harus menentuka parameter algoritma genetika, memilih representasi yang tepat sesuai

permasalahan, memilih metode reproduksi yang tepat baik crossover maupun mutasi, serta metode seleksi yang dirasa paling cocok dengan representasi.

2.4.1 Parameter Algoritma Genetika

Menentukan parameter algoritma genetika seperti *mutation rate* (mr), *crossover rate* (cr), dan jumlah populasi merupakan proses yang bisa dikatakan tidak mudah mengingat banyaknya kemungkinan pada algoritma genetika. Menentukan jumlah populasi sering kali menjadi permasalahan untuk sebagian peneliti. Semakin besar ukuran populasi akan semakin mudah algoritma untuk mendapatkan hasil yang bervariasi. Namun hal tersebut juga akan menambah waktu, *memory*, dan juga biaya pencarian algoritma genetika. Kebanyakan penelitian menetapkan jumlah populasi di sekitar angka 100, namun jumlah ini selalu bisa dirubah tergantung dengan waktu dan *memory* yang tersedia (Mitchell, 1999).

Tidak ada cara yang baku dalam menentukan *mutation rate* dan *crossover rate*, namun sebagian besar peneliti menggunakan parameter yang digunakan oleh para peneliti sebelumnya. Untuk itu setiap nilai cr dan mr memiliki hasil yang berbeda bergantung pada keberagaman permasalahan dan cara merepresentasikan solusi untuk permasalahan tersebut. Terlebih lagi nilai optimal untuk setiap parameter selalu berubah pada setiap iterasi. Untuk itu dalam penentuan parameter algoritma peneliti seringkali melakukan pengujian untuk mendapatkan parameter yang sesuai dengan objek penelitiannya (Mitchell, 1999).

2.4.2 Representasi Kromosom

Dalam menggunakan algoritma genetika diperlukan sebuah representasi untuk membangun sebuah kromosom menjadi sebuah solusi. Untuk merepresentasikan solusi dalam sebuah kromosom dapat dilakukan dengan berbagai cara. Berikut adalah beberapa contoh representasi yang banyak digunakan oleh para peneliti (Rothlauf, 2006):

2.4.2.1 Binary Representations

Representasi biner adalah model representasi yang sering digunakan untuk permasalahan klasifikasi. Kromosom pada representasi biner dilambangkan dengan angka biner. Dalam perhitungannya kromosom harus diubah menjadi bilangan desimal agar bisa dilakukan perhitungan fitnessnya. Namun untuk proses reproduksi bisa dilakukan dalam formasi biner (Rothlauf, 2006).

2.4.2.2 Integer Representations

Representasi integer biasanya digunakan dengan representasi permutasi dan kebanyakan para peneliti menggunakan representasi integer dan permutasi untuk permasalahan mengenai optimasi yang melibatkan kombinasi (Gen & Cheng, 2000).

2.4.2.3 Real-coded Representations

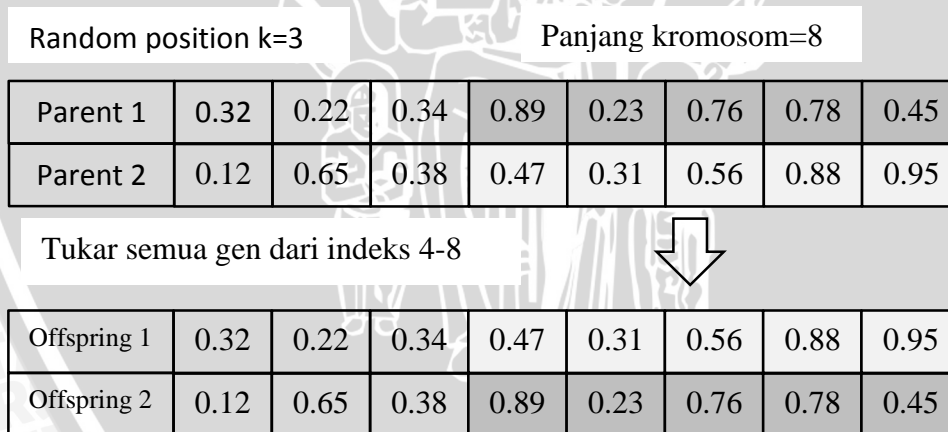
Representasi real-coded algoritma genetika seringkali digunakan karena dapat mencakup bilangan dalam skala yang cukup besar dan range pencarian yang lebih lebar. Model representasi ini lebih banyak dipilih oleh para peneliti karena sangat memudahkan dalam implementasinya ke dalam proses komputasinya. Solusi yang dihasilkan dari metode representasi ini memiliki presisi yang lebih tinggi dibandingkan dengan representasi biner (Magalhaes-Mendes, 2013).

2.4.3 Crossover

Proses *crossover* merupakan proses menggabungkan dua informasi genetik dari dua individu untuk mendapatkan satu atau lebih individu baru. Banyak metode telah dikenalkan oleh para peneliti dan telah banyak digunakan untuk menyelesaikan berbagai macam permasalahan. Berikut adalah beberapa metode *crossover* yang banyak digunakan dalam penelitian.

2.4.3.1 Single-point Crossover

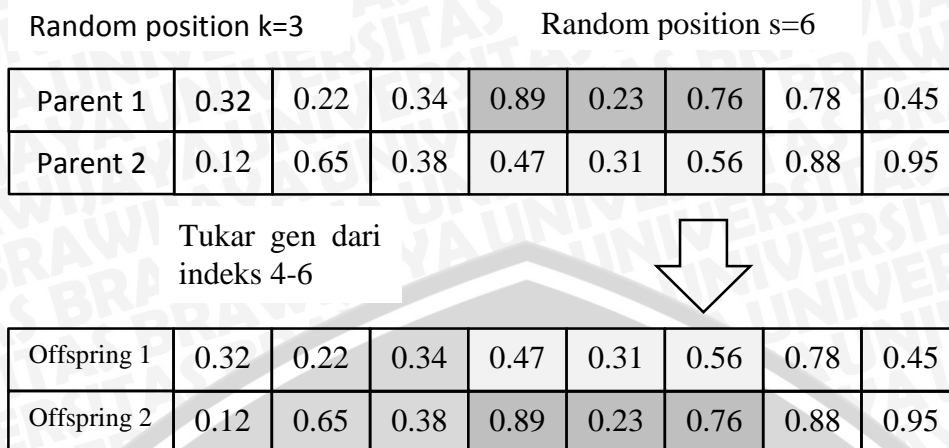
Metode *single-point crossover* ini sering digunakan dalam penelitian karena kemudahannya. Pertama dipilih dua kromosom secara acak dari populasi *parents*. Kemudian pilih posisi secara acak mulai dari 1 hingga panjang kromosom tersebut. Kemudian bentuk individu baru dengan menukar kedua sisi diantara titik potong (Magalhaes-Mendes, 2013). Ilustrasi proses *single-point crossover* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi *Single-point Crossover*

2.4.3.2 Two-point Crossover

Metode ini hampir sama dengan metode *single-point crossover*. Namun pada metode ini terdapat dua titik potong yang dipilih secara acak. Kemudian dibuatlah individu baru dengan menukar *range* titik potong antar *parents* (Magalhaes-Mendes, 2013). Ilustrasi proses *two-point crossover* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Ilustrasi Two-point Crossover

2.4.3.3 Extended Intermediate Crossover

Metode *crossover* ini dapat menjangkau titik manapun dibandingkan metode *crossover* lainnya. Dalam metode ini menggunakan nilai alfa (α) dimana nilainya acak untuk setiap gennya. Namun nilai tersebut akan tetap untuk setiap individunya. Detail perhitungan metode ini dapat dilihat pada persamaan 2.2 (Kaya, et al., 2012).

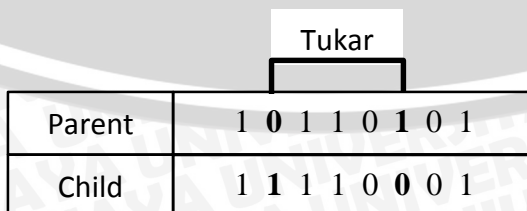
$$offspring = parent1 + random(parent2 - parent1) \quad (2.2)$$

2.4.4 Mutasi

Proses mutasi merupakan proses untuk menghasilkan *offspring* baru dari satu individu *parents*. Jika proses *crossover* bertujuan untuk mencari solusi terbaik dari solusi yang telah ada, maka proses mutasi dapat dikatakan bertujuan untuk membantu menjelajahi ruang lingkup pencarian itu sendiri. Banyak sekali metode mutasi yang dikenalkan oleh para peneliti dan digunakan pada berbagai macam representasi.

2.4.4.1 Interchanging

Pada metode mutasi ini, dipilih satu individu *parents*. Kemudian dipilih dua gen sepanjang kromosom secara acak. Untuk mendapatkan individu baru dilakukan pertukaran antara kedua gen tersebut. Ilustrasi *interchanging mutation* bisa dilihat pada gambar 2.6 (Sivanandam & Deepa, 2008).



Gambar 2.6 Ilustrasi Interchanging Mutation

2.4.4.2 Random Exchange Mutation

Pada metode mutasi ini, hanya satu gen yang akan diubah nilainya. Dipilih satu individu *parent* untuk proses mutasi, kemudian dipilih secara acak gen mana yang akan dimutasi pada individu tersebut. Untuk melakukan metode ini, gen yang terpilih akan ditambahkan / dikurangkan dengan suatu bilangan acak yang bernilai kecil. Namun bilangan acak ini harus dikalikan terlebih dahulu dengan selisih nilai gen terbesar dan terkecil pada variabel yang sama dalam populasi *offspring*. Proses tersebut bisa digambarkan pada persamaan 2.3 (Mahmudy, 2013).

$$x_i = x_i + r(\max_i - \min_i) \quad (2.3)$$

Dimana r adalah bilangan acak yang bernilai kecil, misalkan nilai r merupakan bilangan acak pada interval tertentu.

2.4.5 Seleksi

Prinsip dari algoritma genetika yakni seleksi alam, dimana proses seleksi akan menjadi perantara algoritma dalam menemukan solusi optimalnya. Selama dua dekade terakhir banyak metode seleksi yang dikenalkan oleh para peneliti.

2.4.5.1 Roulette Wheel Selection

Roulette wheel selection pertama kali dikenalkan oleh Holland dan merupakan metode seleksi yang paling banyak dikenal. Konsep dasar dari metode seleksi ini meniru pada permainan *spinning wheel* dimana proses seleksi tersebut akan mengacak kromosom mana yang akan masuk pada populasi berikutnya. Pengacakan dilakukan sampai jumlah populasi selanjutnya terpenuhi (Gen & Cheng, 2000).

2.4.5.2 Elitism Selection

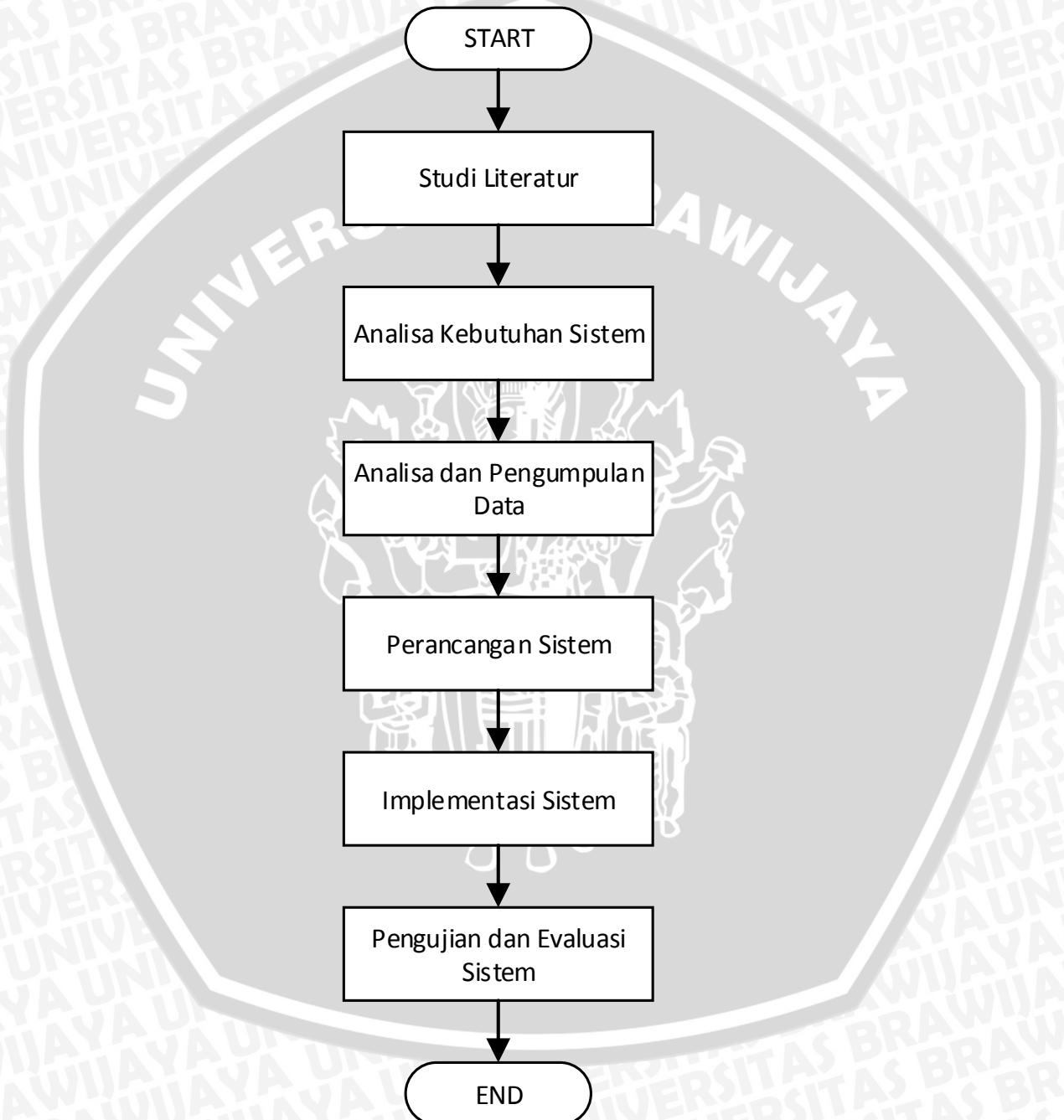
Seleksi elitism hanya akan memilih individu dengan fitness terbaik dari gabungan populasi *parents* dan *offspring*. Seleksi ini banyak digunakan untuk permasalahan optimasi karena kemungkinan untuk mendapatkan kromosom yang sama adalah nol dibandingkan dengan metode *roulette wheel*. Seleksi ini akan mengurutkan individu berdasarkan fitnessnya dan mengambil individu dengan fitness terbaik sejumlah ukuran populasi untuk dijadikan *parents* pada generasi berikutnya (Gen & Cheng, 2000).

2.4.5.3 Tournament Selection

Seleksi ini dikenalkan oleh Goldberg, Korb, dan Deb dimana metode ini akan secara acak memilih satu set kromosom kemudian dibandingkan satu sama lain dan dipilih yang terbaik diantaranya. Jumlah kromosom yang terdapat dalam satu set yang terpilih disebut *tournament size*. *Tournament size* yang biasa digunakan oleh para peneliti yakni 2, untuk itu metode seleksi ini sering disebut *binary tournament selection* (Gen & Cheng, 2000).

BAB 3 METODOLOGI

Pada bab ini akan dijelaskan tahapan penelitian yang akan dilakukan dan kebutuhan sistem dalam membangun penelitian guna menyelesaikan permasalahan menggunakan algoritma genetika. Tahapan metodologi penelitian pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi literatur yang dilakukan guna membangun penelitian ini dilakukan dengan cara mengumpulkan dan mempelajari berbagai macam literatur yang berhubungan dengan penelitian ini diantaranya yang berhubungan dengan kebutuhan nutrisi sapi perah dan algoritma genetika. Literatur yang digunakan dalam menunjang penelitian ini bersumber dari jurnal-jurnal nasional maupun internasional, *paper conference* nasional maupun internasional, laporan penelitian lain, dan buku literatur mengenai teori terkait.

3.2 Analisa Kebutuhan Sistem

Analisa kebutuhan ini bertujuan untuk menganalisis semua kebutuhan yang diperlukan sebelum pembuatan sistem dalam penelitian optimasi komposisi pakan sapi perah menggunakan algoritma genetika. Analisis kebutuhan sistem disesuaikan dengan lokasi, variabel, dan kebutuhan penelitian. Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem ini meliputi :

1. Kebutuhan Hardware meliputi :
 - *Personal Computer* (PC) / Laptop / *Notebook*
2. Kebutuhan Software meliputi :
 - Microsoft Windows 7 atau 8 sebagai sistem operasi.
 - Netbeans 8.02 sebagai *compiler* dalam pembuatan sistem menggunakan bahasa pemrograman Java.
 - Microsoft Visio 2013 yang akan membantu dalam pembuatan diagram.

3.3 Analisa dan Pengumpulan Data

Data yang digunakan pada penelitian ini diambil dari beberapa sumber. Data-data ini nantinya akan digunakan dalam penelitian ini baik dalam perhitungan maupun untuk analisa hasil optimasi. Data yang dikumpulkan untuk pelaksanaan penelitian ini meliputi :

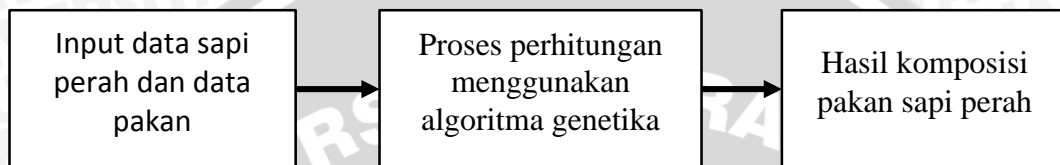
- Data jenis pakan yang biasa digunakan untuk ternak dari peternak di kota Batu beserta berat badan sapi.
- Data pemberian pakan harian dan produksi susu sapi harian dari peternak di kota Batu.
- Data harga jenis pakan dan kandungan lemak serta kualitas susu dari KUD Batu
- Data kebutuhan nutrisi sapi perah dari NRC 1978 dan data kandungan nutrisi pakan dari NRC 2001 dan Fakultas Peternakan

3.4 Perancangan Sistem

Perancangan sistem dibangun berdasarkan hasil analisis kebutuhan dan pengambilan data. Perancangan ini dibuat untuk mempermudah dalam implementasi sistem. Sistem ini akan dibangun menggunakan bahasa pemrograman Java.

1. Alur kerja sistem

Dalam tahap ini perancangan yang dibuat yakni mengenai jalannya sistem dalam menyelesaikan permasalahan. Alur jalannya sistem secara umum dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Kerja Sistem

2. Perancangan antarmuka

Perancangan antarmuka ini digunakan untuk mempermudah dalam mengimplementasikan sistem yang akan dibangun.

3. Perancangan pengujian

Perancangan pengujian sistem meliputi :

- Pengujian dalam menentukan jumlah populasi terbaik untuk proses algoritma sistem.
- Pengujian dalam menentukan *crossover* dan *mutation rate* yang paling baik untuk proses algoritma sistem.
- Pengujian dalam menentukan jumlah generasi yang paling baik untuk proses algoritma.
- Pengujian dalam menentukan interval gen yang tepat untuk proses algoritma sistem.

3.5 Implementasi Sistem

Implementasi sistem adalah tahap membangun sistem sesuai dengan perancangan yang telah dibuat dan dengan menerapkan hal yang telah didapatkan pada proses studi literatur. Sistem ini akan dibangun dengan Java sebagai *platform* pengembangnya. Tahap implementasi yang akan dilakukan meliputi implementasi *interface*, implementasi algoritma genetika, dan implementasi pengujian. Keluaran yang dihasilkan dari proses implementasi ini yakni sistem optimasi pakan sapi perah menggunakan algoritma genetika.

3.6 Pengujian dan Evaluasi Sistem

Proses pengujian dan evaluasi sistem dilakukan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan spesifikasi yang telah ditetapkan. Proses pengujian dan evaluasi ini dilakukan guna menetapkan parameter algoritma genetika yang paling baik mulai dari mencari *crossover* dan *mutation rate*, menetapkan jumlah populasi, menetapkan jumlah generasi, dan menetapkan interval gen yang tepat. Pengujian ini dilakukan dengan menjalankan sistem berulang kali dengan nilai parameter yang berbeda. Kemudian dilakukan proses evaluasi untuk membandingkan hasil optimasi keluaran sistem dengan kasus yang ada di lapangan.

3.7 Pengambilan Keputusan

Proses pengambilan keputusan akan dilakukan setelah semua proses yang meliputi perancangan, implementasi, dan pengujian selesai. Penarikan kesimpulan dilakukan berdasarkan hasil pengujian yang telah dilakukan dan analisa global. Penarikan kesimpulan dilakukan guna menjawab rumusan masalah yang telah dilakukan sebelumnya. Setelah itu dilakukan penulisan saran bagi penelitian yang sedang dilakukan. Saran tersebut ditulis untuk memperbaiki atau menyempurnakan penelitian yang sedang dilakukan serta dijelaskan apa yang masih kurang dari penelitian sehingga dapat dikembangkan pada penelitian selanjutnya.



BAB 4 PERANCANGAN

Bab ini menjelaskan perancangan penyelesaian masalah menggunakan algoritma genetika dimulai dari proses inialisasi, reproduksi, dan evaluasi, perancangan pengujian algoritma genetika, dan perancangan antarmuka sistem.

4.1 Formulasi Permasalahan

Permasalahan yang akan diselesaikan yakni pembentukan komposisi pakan sapi yang tepat. Untuk dapat menyelesaikan permasalahan ini dibutuhkan beberapa variabel untuk mendapatkan komposisi pakan yang tepat. Variabel-variabel ini meliputi berat badan sapi, rata-rata produksi susu sapi per ekor per hari, kandungan lemak dalam susu sapi, minggu laktasi sapi, dan juga jenis bahan pakan yang *diinputkan* oleh *user*. Solusi yang nanti akan diberikan oleh sistem berupa komposisi dari jenis pakan pilihan *user*. Data sapi yang diperoleh dari *user* digunakan untuk menghitung kebutuhan nutrisi hidup pokok dan kebutuhan nutrisi produksi sapi perah. Kemudian dari beberapa jenis pakan yang didapat dari *user* pula dihitung kandungan nutrisi yang ada didalamnya yang nantinya akan menjadi ketersediaan nutrisi bagi sapi. Setelah perhitungan kebutuhan nutrisi dan ketersediaan didapatkan keduanya dilihat apakah nutrisi pada pakan sudah memenuhi kebutuhan nutrisi sapi. Kekurangan nutrisi pada perhitungan akan dijadikan sebagai *penalty* dan kelebihan ketersediaan nutrisi nantinya akan dihitung sebagai estimasi susu yang dapat dihasilkan oleh sapi. Setelah perhitungan nutrisi selesai dilakukan maka selanjutnya digunakan algoritma genetika untuk menentukan komposisi dengan estimasi produksi maksimal dan harga pakan seminimal mungkin.

Data yang diperoleh dari para peternak sapi perah di Batu didapatkan 10 jenis bahan pakan yang digunakan oleh para peternak. Karena kandungan nutrisi bahan pakan yang diketahui hanya kandungan BK, PK, dan TDN, maka dibutuhkan kandungan NE_L untuk mengetahui kandungan energi dalam setiap jenis pakan. Dengan itu NRC (2001) telah memberikan persamaan yang dapat digunakan untuk menghitung kandungan NE_L dalam pakan melalui kandungan TDN yang dapat dihitung dengan persamaan 4-1.

$$NE_L(Mcal/kg) = 0.0245 \times TDN(\%) - 0.12 \quad (4-1)$$

Daftar jenis pakan beserta kandungan nutrisi dan harga pakan per kilogram dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel Kandungan Nutrisi Bahan Pakan Sapi Perah

Jenis Pakan	BK(%)	PK(%)	TDN(%)	NEL (Mkal/kg)	Harga (Rp)
Rumput Gajah	36.5	12.8	55.7	1.24465	1250
Tebon Jagung	35.1	8.8	68.8	1.5656	1700
Pollard	82.82	18.12	84.78	1.95711	4500
Bapro	81.79	18.69	32.456	0.675172	3000
Ampas Tahu	20.66	25.96	73.625	1.683813	500

Jenis Pakan	BK(%)	PK(%)	TDN(%)	NEL (Mkal/kg)	Harga (Rp)
Limbah Roti	68.3	15	89.3	2.06785	250
Jerami Padi	80.8	3.9	45.47	0.994015	500
Kulit Kedelai	90.9	13.9	67.3	1.52885	250
Daun Lamtoro	30	32.12	69.017	1.570917	2000
Bekatul	90.6	15.5	84.8	1.9576	2500

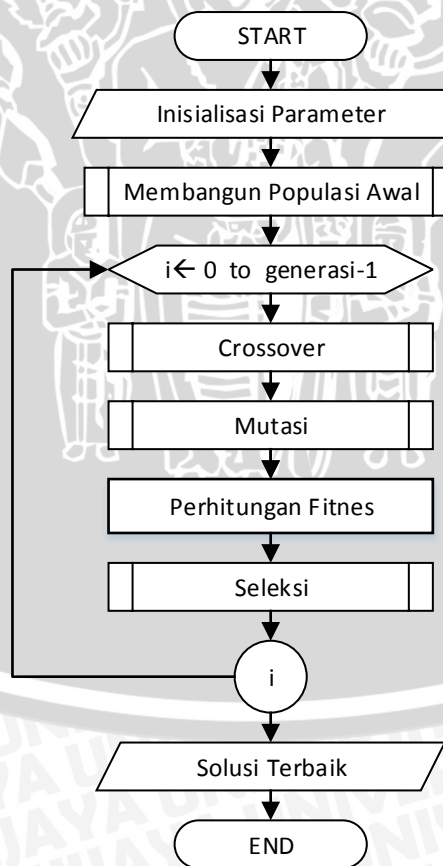
Sumber : (Council, 2001) (Wijaya, 2015)

Adapun formulasi masalah yang bisa dicontohkan adalah sebagai berikut :

Seorang peternak memiliki seekor sapi perah dengan bobot sapi 400 kg. Sapi tersebut menghasilkan 11 liter susu setiap harinya dengan kandungan lemak dalam susu tersebut sebesar 4%. Peternak tersebut hanya memiliki bahan pakan rumput gajah, tebon jagung, dan bekatul.

4.2 Alur Penyelesaian Menggunakan Algoritma Genetika

Pada sub bab ini akan dijelaskan siklus penyelesaian permasalahan dengan algoritma genetika. Dalam satu siklus algoritma genetika terdapat lima proses utama untuk membentuk satu generasi baru. Berdasarkan formulasi masalah yang telah dijelaskan pada sub bab 4.1, alur penyelesaian dengan algoritma genetika dapat dilihat pada Gambar 4.1.



Gambar 4.1 Flowchart Penyelesaian Algoritma Genetika

4.2.1 Membangun Individu dan Menentukan *Fitness*

Tahap awal sebelum memasuki siklus algoritma genetika yakni menentukan rangkaian sebuah kromosom dan cara menghitung fitnessnya. Dalam permasalahan ini panjang kromosom akan disesuaikan dengan jumlah jenis pakan yang dipilih oleh peternak. Dan masing-masing gen dalam kromosom tersebut akan berisi bilangan *random* yang menandakan bobot dari jenis pakan tersebut. Gambar 4.2 merupakan contoh sebuah kromosom yang dibangun berdasarkan contoh formulasi permasalahan.

Rumput Gajah	Tebon Jagung	Bekatul
7.7	8.4	6.2

Gambar 4.2 Contoh Representasi Kromosom

Untuk menghitung *fitness* kromosom tersebut diperlukan perhitungan kebutuhan nutrisi sapi perah dan ketersediaan nutrisi dari jenis pakan tersebut. Dalam perhitungan *fitness* ini digunakan *penalty* yakni apabila ketersediaan nutrisi dalam pakan tidak mencukupi kebutuhan nutrisi sapi perah.

4.2.1.1 Menghitung Kebutuhan Nutrisi Sapi Perah

Dalam menghitung kebutuhan nutrisi sapi perah, terdapat dua jenis kebutuhan yang harus dihitung yakni kebutuhan nutrisi hidup pokok dan kebutuhan nutrisi produksi. Kebutuhan nutrisi sapi perah yang harus dihitung yakni meliputi BK, PK, TDN, dan NE_L .

a. Kebutuhan Hidup Pokok

Dari Tabel 2.3 dapat dihitung kebutuhan bahwa sapi dengan bobot 400 kg membutuhkan :

$$NE_L = 7.16 \text{ Mkal}$$

$$TDN = 3.15 \text{ kg}$$

$$PK = 0.373 \text{ kg}$$

b. Kebutuhan Produksi

Dari Tabel 2.4 dapat diketahui kebutuhan nutrisi sapi untuk memproduksi satu liter susu. Untuk menghitung keseluruhan produksi sebagai berikut :

$$NE_L = 0.74 \times 11 \text{ liter} = 8.14$$

$$TDN = 0.326 \times 11 \text{ liter} = 3.586$$

$$PK = 0.087 \times 11 \text{ liter} = 0.957$$

c. Kebutuhan Nutrisi Total

Kebutuhan total nutrisi sapi perah dihitung dengan menjumlahkan kebutuhan hidup pokok dan produksi.

$$NE_L(\text{aktivitas})10\% \text{ HP} = 10\% \times 7.16 = 0.716$$

$$NE_L total = 7.16 + 8.14 + 0.716 = 16.016$$

$$TDN total = 3.15 + 3.586 = 6.736$$

$$PK total = 0.373 + 0.957 = 1.33$$

$$BK total = 16.016 \div 1.52 = 10.5368$$

4.2.1.2 Menghitung Ketersediaan Nutrisi Pakan

Dalam menghitung ketersediaan nutrisi pakan, dicari nutrisi pakan sesuai dengan Tabel 4.1 sesuai dengan jenis pakan. Nilai nutrisi tersebut kemudian dikalikan dengan bobot masing-masing jenis pakan yang ada pada kromosom.

a. Rumpun Gajah

$$BK = 36.5\% \times 7.7 = 2.8105$$

$$PK = 12.8\% \times 7.7 = 0.9856$$

$$TDN = 55.7\% \times 7.7 = 4.2889$$

$$NE_L = 1.24465 \times 7.7 = 9.583805$$

b. Tebon Jagung

$$BK = 35.1\% \times 8.4 = 2.9484$$

$$PK = 8.8\% \times 8.4 = 0.7392$$

$$TDN = 68.8\% \times 8.4 = 5.7792$$

$$NE_L = 1.5656 \times 8.4 = 13.15104$$

c. Bekatul

$$BK = 90.6\% \times 6.2 = 5.6172$$

$$PK = 15.5\% \times 6.2 = 0.961$$

$$TDN = 84.8\% \times 6.2 = 5.2576$$

$$NE_L = 1.9576 \times 6.2 = 12.13712$$

d. Total ketersediaan nutrisi pakan

$$BK = 2.8105 + 2.9484 + 5.6172 = 11.3761$$

$$PK = 0.9856 + 0.7392 + 0.961 = 2.6858$$

$$TDN = 4.2889 + 5.7792 + 5.2576 = 15.3257$$

$$NE_L = 9.583805 + 13.15104 + 12.13712 = 34.871965$$

4.2.1.3 Menghitung Fitness

Setelah diketahui semua kebutuhan nutrisi sapi perah dan ketersediaan nutrisi dalam pakan, selanjutnya dilakukan perhitungan *fitness*. Perhitungan *fitness* melibatkan nilai *penalty* dimana nilai *penalty* akan berlaku jika ketersediaan nutrisi dalam pakan kurang dari kebutuhan nutrisi sapi perah, dan nilai *penalty* 0 jika kebutuhan sudah tercukupi. Nilai *penalty* tersebut dihitung berdasarkan jenis nutrisinya kemudian dijumlahkan menjadi nilai *penalty* total.

Tabel 4.2 Tabel Perbandingan Nutrisi Sapi dan Pakan

	Total Kebutuhan	Total Ketersediaan	Penalty	Kelebihan Nutrisi
BK (kg)	10.536842	11.3761	0	0.839257
PK (kg)	1.33	2.6858	0	1.3558
TDN (kg)	6.736	15.3257	0	8.5897
NEL (Mkal/Kg)	16.016	34.87197	0	18.855965
Total			0	

Dari kelebihan nutrisi pakan dapat dihitung perkiraan produksi susu yang dapat dihasilkan dengan membagi masing-masing nilai kelebihan nutrisi dengan kebutuhan nutrisi produksi pada Tabel 2.4.

$$Prod. menurut BK = 0.839257895 \div 1.57 = 0.534073$$

$$Prod. menurut PK = 1.3558 \div 0.087 = 15.58391$$

$$Prod. menurut TDN = 8.5897 \div 0.326 = 26.34877$$

$$Prod. menurut NEL = 18.855965 \div 0.74 = 25.48103$$

Dari hasil perhitungan perkiraan produksi melalui keempat nutrisi tersebut diambil hasil terkecil sebagai hasil akhir perkiraan produksi. Selanjutnya dilakukan perhitungan total harga pakan dari sebuah kromosom dengan mengalikan harga masing-masing jenis pakan dengan bobot tiap jenis pakan.

$$Rumput Gajah = 7.7 \times Rp 1250 = Rp 9.625$$

$$Tebon Jagung = 8.4 \times Rp 1700 = Rp 14.280$$

$$Bekatul = 6.2 \times Rp 2500 = Rp 15.500$$

$$Total Harga = Rp 9.625 + Rp 14.280 + Rp 15.500 = Rp 39.405$$

Selanjutnya dapat dihitung nilai *fitness* dari kromosom tersebut dengan menggunakan persamaan 4.2.

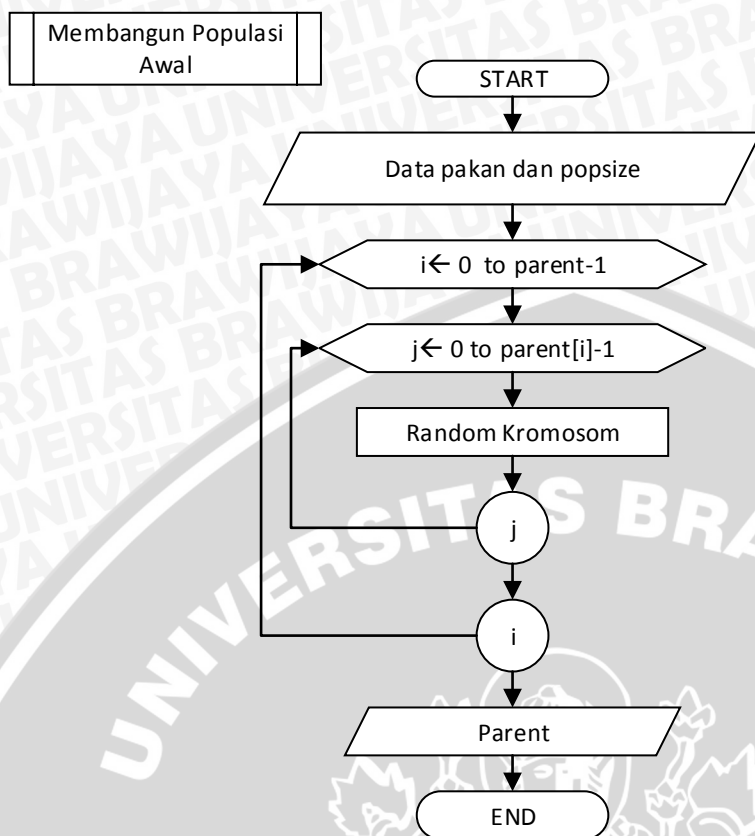
$$Fitnes = produksi + \frac{1}{harga} + \frac{1}{penalty+1} \tag{4.2}$$

$$Fitnes = 0.534073 + \frac{1}{39405} + \frac{1}{0+1} = 1.5340985$$

4.2.2 Menentukan Populasi Awal

Dalam penyelesaian algoritma genetika diperlukan sekumpulan individu awal yang nantinya akan dikembangkan dengan cara reproduksi. Populasi awal ini dibangkitkan dengan secara acak sesuai dengan representasi kromosomnya. Jumlah kromosom yang dibangkitkan pada populasi awal disesuaikan dengan nilai *popsiz*e yang telah ditentukan. Panjang kromosom akan menyesuaikan dengan banyaknya jenis pakan yang dipilih oleh *user*. Sedangkan nilai dari setiap gen akan dipilih secara acak dengan interval 1-10. Gambar 4.3 menjelaskan proses membangun populasi awal.





Gambar 4.3 Flowchart Proses Pembentukan Populasi Awal

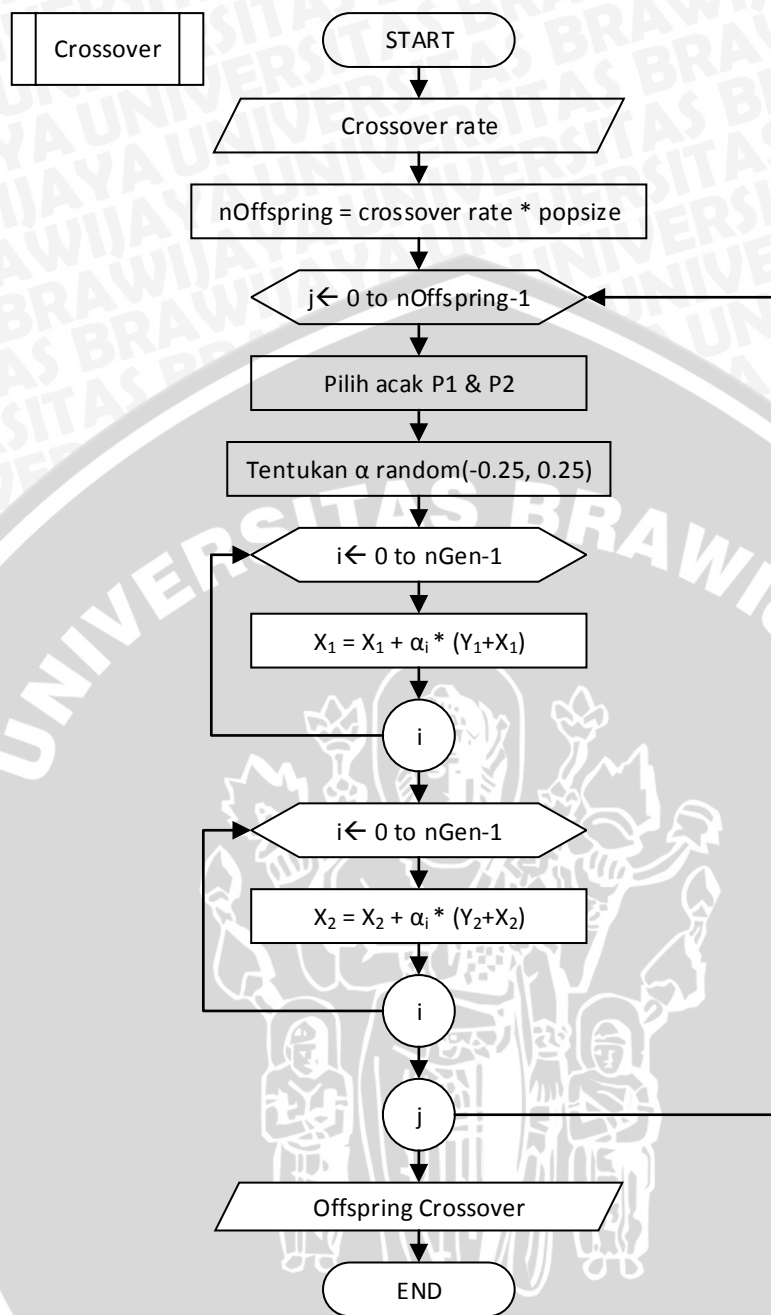
Dari proses diatas dapat dibentuk individu sejumlah dengan populasi yang ditentukan. Tabel 4.3 merupakan contoh sebuah populasi yang terdiri dari lima kromosom sesuai dengan formulasi permasalahan.

Tabel 4.3 Inisialisasi Populasi Awal

	Tebon Jagung	Jerami Padi	Bekatul
P1	7.7	8.4	6.2
P2	8.4	9.7	4.6
P3	9.98	6.31	3
P4	10	4.1	1
P5	7	9	3

4.2.3 Crossover

Proses selanjutnya setelah didapatkan populasi awal yakni memulai proses reproduksi. Reproduksi yang pertama dilakukan yakni menggunakan proses *crossover* dimana akan dipilih dua individu *parents* secara acak untuk dimodifikasi gennya agar membentuk individu baru. Dalam penelitian ini metode *crossover* yang digunakan adalah *extended intermediate crossover*. Gambar 4.4 merupakan *flowchart* dari proses *crossover*.



Gambar 4.4 Flowchart Proses Crossover

Jumlah *offspring* yang dihasilkan dari proses *crossover* ini bergantung dengan nilai *crossover rate*, yang jumlahnya dapat dihitung dengan mengalikan nilai *crossover rate* dan *popsize*.

Pada formulasi permasalahan diketahui nilai *cr* adalah 0.4 dengan *popsize* 5. Maka hasil *offspring* dari proses *crossover* adalah $0.4 \times 5 = 2$ *offspring*. Dari sebuah proses *crossover* akan menghasilkan dua *offspring* yang nantinya akan dipilih salah satu *offspring* dengan *fitness* tertinggi sebagai *offspring* utama.

- **Crossover 1**

Pada proses ini nilai $\alpha_1 = 0.21$, $\alpha_2 = 0.087$, $\alpha_3 = -0.16$ dengan *parent* yang terpilih yakni P2 dan P5

Tabel 4.4 Kromosom Terpilih Untuk Crossover

P2	8.4	5.9	4.6
P5	7	9	3

Hitung tiap gen untuk *offspring 1*

$$x_1 = 8.4 + 0.21(7 - 8.4) = 8.106$$

$$x_2 = 5.9 + 0.087(9 - 5.9) = 6.1697$$

$$x_3 = 4.6 - 0.16(3 - 4.6) = 4.856$$

Hitung tiap gen untuk *offspring 2*

$$x_1 = 7 + 0.21(8.4 - 7) = 7.294$$

$$x_2 = 9 + 0.087(5.9 - 9) = 8.7303$$

$$x_3 = 3 - 0.16(4.6 - 3) = 2.744$$

Tabel 4.4 merupakan dua kromosom *parent* yang terpilih untuk proses *crossover*, sedangkan Tabel 4.5 merupakan kromosom hasil proses *crossover* dengan C1 terpilih sebagai *offspring* yang akan menjadi hasil akhir dari proses *crossover* pertama.

Tabel 4.5 Kromosom Hasil Crossover

	X1	X2	X3	Fitness
C1	8.106	6.1697	4.856	0.496789
C2	7.294	8.7303	2.744	0.300863

- **Crossover 2**

Pada proses ini nilai $\alpha_1 = 0.101$, $\alpha_2 = -0.176$, $\alpha_3 = 0.55$ dengan *parent* yang terpilih yakni P3 dan P5

Tabel 4.6 Kromosom Terpilih Untuk Crossover

P3	9.87	6.31	3
P5	7	9	3

Hitung tiap gen untuk *offspring 1*

$$x_1 = 9.87 + 0.101(7 - 9.87) = 9.67902$$

$$x_2 = 6.31 - 0.176(9 - 6.31) = 5.83656$$

$$x_3 = 3.1 + 0.55(3 - 3) = 3$$

Hitung tiap gen untuk *offspring 2*

$$x_1 = 7 + 0.101(9.87 - 7) = 7.30098$$

$$x_2 = 9 - 0.176(6.31 - 9) = 9.47344$$

$$x_3 = 3 + 0.55(3 - 3) = 3$$

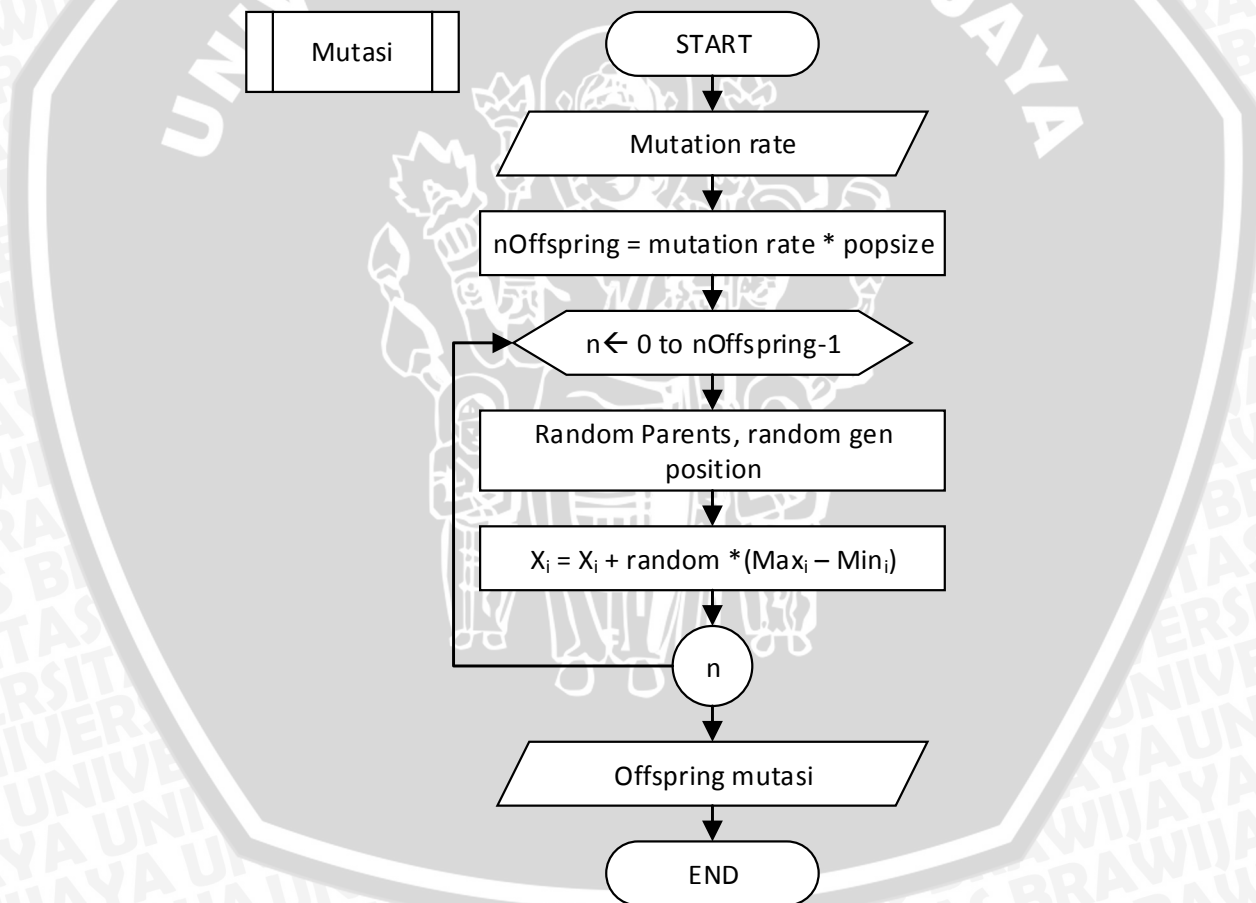
Tabel 4.6 merupakan dua kromosom *parent* yang terpilih untuk proses *crossover* sedangkan Tabel 4.7 merupakan kromosom hasil proses *crossover* dengan C2 terpilih sebagai *offspring* yang akan menjadi hasil akhir dari proses *crossover* kedua.

Tabel 4.7 Kromosom Hasil Crossover

	X1	X2	X3	Fitness
C1	9.67902	5.83656	3	0.2366638
C2	7.30098	9.47344	3	0.543480056

4.2.4 Mutasi

Proses mutasi dilakukan untuk mendapatkan individu baru hanya dengan memodifikasi satu kromosom. Metode mutasi yang digunakan adalah *random exchange mutation*. Gambar 4.5 merupakan *flowchart* proses mutasi.



Gambar 4.5 Flowchart Proses Mutasi

Offspring hasil proses mutasi bergantung pada nilai *mutation ratenya*. Jumlah *offspring* dari proses ini dapat diketahui dengan mengalikan nilai *mutation rate* dengan *popsize*. Dari formulasi permasalahan diketahui bahwa nilai *mr* adalah 0.2 dengan *popsize* 5. Maka dapat dihitung bahwa *offspring* yang dihasilkan dari



proses mutasi sebanyak $0.2 \times 5 = 1$ *offspring*. Dimisalkan nilai r acak bernilai 0.1, individu *parent* yang terpilih adalah P4, dengan gen yang terpilih untuk dimutasi adalah x_2 . Tabel 4.8 merupakan contoh kromosom *parent* yang terpilih untuk proses mutasi. Sedangkan Tabel 4.9 merupakan kromosom hasil proses mutasi.

Tabel 4.8 Kromosom terpilih untuk proses mutasi

P4	10	4.1	1
----	----	-----	---

Lakukan proses mutasi pada gen nomor 2

$$x_2 = 4.1 + 0.1(9.47344 - 6.1697) \quad (4.3)$$

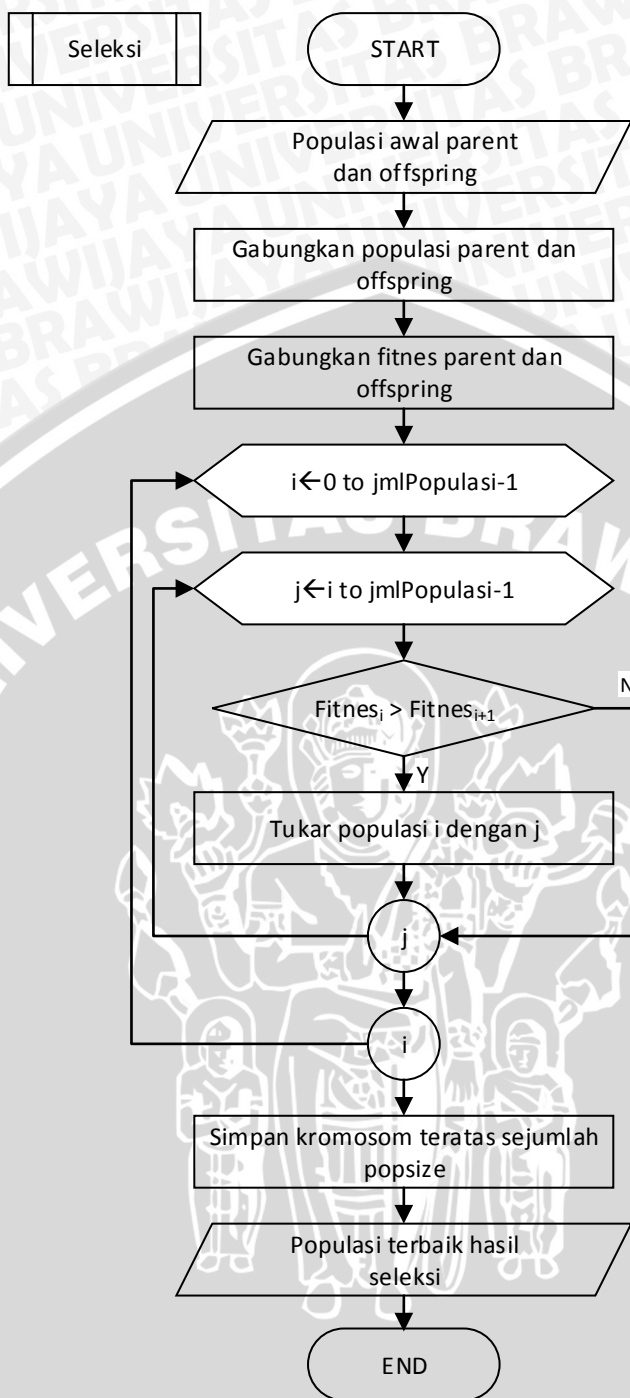
Tabel 4.9 Kromosom hasil proses mutasi

	X1	X2	X3	Fitness
C1	10	4.430374	1	0.184349652

4.2.5 Seleksi

Proses selanjutnya dari algoritma genetika yakni melakukan seleksi terhadap populasi tersebut untuk membentuk sebuah generasi baru. Individu yang diambil dari populasi saat ini akan dijadikan individu *parents* pada generasi selanjutnya. Metode seleksi yang digunakan yakni *elitism selection*. Seleksi ini digunakan karena menjamin bahwa kromosom yang memiliki *fitness* lebih besar akan bertahan hingga generasi selanjutnya. Gambar 4.6 menjelaskan *flowchart* dalam proses seleksi.





Gambar 4.6 Flowchart Proses Seleksi

Dari proses seleksi akan dipilih individu dengan *fitness* terbaik sejumlah *popsize* yang kemudian akan dijadikan pada generasi berikutnya. Dari proses *crossover* dan mutasi telah didapatkan tiga *offspring*, sehingga populasi kromosom saat ini menjadi delapan individu yang terdiri dari lima *parent* dan tiga *offspring*.

Tabel 4.10 Individu total dalam sebuah populasi

	R. Gajah	T. Jagung	Bekatul	<i>Fitness</i>
P1	7.7	8.4	6.2	1.534098583

	R. Gajah	T. Jagung	Bekatul	<i>Fitness</i>
P2	8.4	9.7	4.6	1.060900718
P3	9.98	6.31	3	0.337722072
P4	10	4.1	1	0.180494188
P5	7	9	3	0.322109266
C1	8.106	6.1697	4.856	0.496788827
C2	10.0098	9.47344	3	0.543480056
C3	10	4.430374	1	0.184349652

Dari seluruh populasi *parent* dan *child* pada Tabel 4.10 kemudian dilakukan seleksi *elitism* yakni dengan mengurutkan individu dari *fitness* tertinggi dan memilih lima individu dengan *fitness* tertinggi untuk generasi selanjutnya.

Tabel 4.11 Individu dengan *fitness* tertinggi.

	R. Gajah	T. Jagung	Bekatul	<i>Fitness</i>
P1	7.7	8.4	6.2	1.534098583
P2	8.4	9.7	4.6	1.060900718
P4	10.0098	9.47344	3	0.543480056
C3	8.106	6.1697	4.856	0.496788827
C2	9.98	6.31	3	0.337722072
C1	7	9	3	0.322109266
P3	10	4.430374	1	0.184349652
P5	10	4.1	1	0.180494188

Dari hasil seleksi pada Tabel 4.11 dapat disimpulkan bahwa solusi yang mendekati optimal dapat diambil dari P1 dengan komposisi pakan 7.7 kg rumput gajah, 8.4 kg tebon jagung, dan 6.2 kg bekatul dengan harga total pakan Rp 39.405,00 dan perkiraan pertambahan produksi sebesar 0.503 liter susu.

4.3 Perancangan Pengujian Algoritma Genetika

Pengujian yang akan dilakukan pada penelitian terdiri dari beberapa pengujian. Hal yang akan diuji pada algoritma adalah parameter dari algoritma genetika. Pengujian ini bertujuan untuk menemukan parameter algoritma genetika yang tepat bagi permasalahan yang sedang diselesaikan. Pengujian tersebut terdiri dari pengujian *popsize*, pengujian nilai *crossover rate* dan *mutation rate*, dan pengujian generasi. Parameter objek yang akan digunakan dalam pengujian ini yakni bobot sapi perah 400 kg, produksi susu 11 liter, dan kandungan lemak 4%. Dengan bahan pakan yang digunakan adalah rumput gajah, tebon jagung, dan bekatul.

4.3.1 Pengujian Interval Gen

Pengujian ini dilakukan untuk mendapatkan interval bobot gen yang tepat untuk komposisi pakan sapi perah. Parameter algoritma yang digunakan adalah *popsize* 50, jumlah generasi 100, nilai *cr* 0.4, dan nilai *mr* 0.2. Nilai interval gen

yang akan diuji dimulai dari 10 hingga 100 dengan interval kenaikan 10. Setiap pengujian interval akan dilakukan sebanyak 10 kali percobaan dengan membangkitkan kromosom secara acak. Tabel 4.12 merupakan rancangan pengujian nilai *range* interval gen.

Tabel 4.12 Rancangan Pengujian Interval Gen

Interval	Fitness Percobaan ke - <i>i</i>										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
1 – 10											
1 – 20											
1 – 30											
1 – 40											
1 – 50											
1 – 60											
1 – 70											
1 – 80											
1 – 90											
1 - 100											

4.3.2 Pengujian Ukuran Populasi

Pengujian ukuran populasi dilakukan untuk mendapatkan ukuran populasi yang tepat untuk perhitungan algoritma genetika. Parameter algoritma genetika yang digunakan yakni banyak generasi 100, nilai *cr* 0.4, nilai *mr* 0.2, dan interval gen menggunakan hasil interval terbaik dari pengujian interval gen. Ukuran populasi yang diuji mulai dari 10 dengan kelipatan 30 hingga 250. Setiap pengujian ukuran populasi akan dilakukan sebanyak 10 kali dengan membangkitkan kromosom secara acak. Tabel 4.13 merupakan rancangan pengujian ukuran populasi.

Tabel 4.13 Rancangan Pengujian Ukuran Populasi

Popsize	Fitness Percobaan ke - <i>i</i>										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
10											
40											
70											
100											
130											
160											
190											
220											

Popsize	Fitness Percobaan ke - i										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
250											

4.3.3 Pengujian Banyaknya Generasi

Pengujian banyaknya generasi dilakukan untuk menentukan nilai banyaknya generasi yang tepat pada permasalahan yang sedang diselesaikan. Parameter algoritma genetika yang digunakan yakni nilai cr 0.4, nilai 0.2, interval gen dari hasil pengujian interval gen, dan nilai $popsiz$ dari hasil pengujian $popsiz$. Pengujian ini akan dimulai dari generasi 100 dengan kelipatan 100 hingga 1000. Pengujian tersebut akan dilakukan sebanyak 10 kali untuk setiap jumlah generasi dengan membangkitkan kromosom secara acak. Tabel 4.14 adalah tabel rencana pengujian generasi.

Tabel 4.14 Rancangan Pengujian Jumlah Generasi

Generasi	Fitness Percobaan ke - i										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
100											
200											
300											
400											
500											
600											
700											
800											
900											
1000											

4.3.4 Pengujian Nilai *Crossover Rate* dan *Mutation Rate*

Pengujian ini dilakukan untuk mendapatkan nilai *crossover* dan *mutation rate* yang tepat bagi permasalahan yang sedang diselesaikan. Nilai *crossover* dan *mutation rate* akan digunakan untuk menentukan banyaknya *offspring* yang dihasilkan dari setiap proses *crossover* dan mutasi. Dalam pengujian ini nilai *popsiz* yang digunakan adalah nilai *popsiz* terbaik hasil pengujian ukuran populasi, jumlah generasi yang digunakan juga berasal dari jumlah generasi terbaik dari pengujian generasi, dan interval gen yang digunakan adalah hasil dari pengujian interval gen. Setiap kombinasi nilai cr dan mr dilakukan percobaan sebanyak 10 kali dengan membangkitkan kromosom secara random. Tabel 4.15 merupakan rancangan pengujian nilai *crossover* dan *mutation rate*.

Tabel 4.15 Rancangan Pengujian Nilai Cr dan Mr

Cr Mr	Fitness Percobaan ke - i										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
0 1											
0.1 0.9											
0.2 0.8											
0.3 0.7											
0.4 0.6											
0.5 0.5											
0.6 0.4											
0.7 0.3											
0.8 0.2											
0.9 0.1											
1 0											

4.4 Perancangan Antarmuka

Perancangan antarmuka digunakan untuk memberikan gambaran seperti apa sistem yang akan dibangun nantinya. Implementasi antarmuka sistem ini nantinya akan terdiri dari halaman input data sapi dan pakan dan halman hasil.

4.4.1 Rancangan Antarmuka Halaman Input Data

Gambar 4.7 merupakan rancangan antarmuka pada halaman input data. Pada halaman ini terdapat panel-panel untuk memasukkan data sapi, data pakan, dan data kebutuhan nutrisi sapi perah. Keterangan objek yang terdapat pada gambar adalah sebagai berikut :

1. Tab Input Data yang memuat halaman input data
2. Panel input data sapi perah
3. Panel untuk menampilkan kebutuhan sapi berdasarkan data input
4. Tombol tambah bahan pakan
5. Tabel daftar pakan yang dipilih beserta kandungan nutrisinya
6. Tombol untuk memulai proses optimasi

OPTIMASI KOMPOSISI PAKAN SAPI PERAH DENGAN ALGORITMA GENETIKA	
1	
Input Data	Hasil Optimasi
2	4
3	5
	6

Gambar 4.7 Rancangan Antarmuka Halaman Input Data

4.4.2 Rancangan Antarmuka Halaman Hasil Optimasi

Halaman hasil optimasi hanya akan berisi data hasil perhitungan algoritma dan juga hasil optimasi komposisi pakan. Gambar 4.8 merupakan rancangan halaman hasil optimasi yang terdiri dari beberapa panel untuk menampilkan sejumlah kromosom terbaik.

OPTIMASI KOMPOSISI PAKAN SAPI PERAH DENGAN ALGORITMA GENETIKA	
Input Data	Hasil Optimasi 1
2	
3	

Gambar 4.8 Rancangan Antarmuka Halaman Hasil Perhitungan

Keterangan dari rancangan antarmuka pada Gambar 4.8 adalah :

1. Tab Hasi Optimasi
2. Tabel untuk menampilkan hasil kromosom terbaik dari setiap generasi.
3. Panel untuk menampilkan hasil akhir optimasi komposisi pakan.

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem optimasi yang dibangun berdasarkan perancangan sebelumnya. Proses implementasi dilakukan untuk merealisasikan rancangan sistem dan untuk mengetahui apakah sistem yang dibangun telah memenuhi tujuan awal penelitian.

5.1 Implementasi Algoritma Genetika

Pada bab perancangan sebelumnya telah dijelaskan rancangan algoritma dalam penyelesaian masalah optimasi komposisi pakan sapi perah. Dalam hal ini proses algoritma yang akan diimplementasikan terdiri dari 4 proses yang meliputi proses inialisasi populasi awal, proses *crossover*, proses mutasi, dan proses seleksi.

5.1.1 Implementasi Penentuan Populasi Awal

Implementasi penentuan populasi awal merupakan proses pertama dalam algoritma genetika. Proses ini bertujuan untuk membangun sekumpulan individu awal yang akan bertindak sebagai *parent* yang nantinya akan direproduksi. Jumlah individu dalam populasi awal ditentukan oleh nilai *popsize* dan panjang kromosom untuk setiap individu ditentukan oleh banyaknya jumlah pakan. Gen untuk setiap kromosom merupakan bilangan random yang merupakan bobot dari tiap bahan pakan. Implementasi program penentuan populasi awal dapat dilihat pada Gambar 5.1.

```

1 public void initPopulasi() {
2     parent = new double[popSize][nGen];
3     fitnessP = new double[popSize];
4     for (int i = 0; i < parent.length; i++) {
5         for (int j = 0; j < parent[i].length; j++) {
6             parent[i][j] = randomGen();
7         }
8     }
9 }

```

Gambar 5.1 Source Code Penentuan Populasi Awal

Penjelasan *source code* pada Gambar 5.1 tentang penentuan populasi awal adalah sebagai berikut :

- Baris 2-3 : inialisasi array tempat kromosom dan *fitness*
- Baris 4-8 : pembuatan kromosom *parent* baru dengan gen random sejumlah *popsize*

5.1.2 Implementasi Proses *Crossover*

Implementasi proses *crossover* merupakan salah satu proses dalam tahap reproduksi. Proses *crossover* ini menggabungkan dan memodifikasi dua kromosom *parent* sehingga menghasilkan dua kromosom *child*. Proses *crossover* yang digunakan yakni *extended intermediate crossover*. Implementasi program untuk proses *crossover* dapat dilihat pada Gambar 5.2.

```

1 public double[][] crossOver() {
2     int p1 = randomParent();
3     int p2 = randomParent();
4     while (p2 == p1) {
5         p2 = randomParent();
6     }
7     double[] crP1 = parent[p1];
8     double[] crP2 = parent[p2];
9     double[] alfa = getAlfa();
10
11    double[][] offSpr = new double[2][nGen];
12    for (int i = 0; i < nGen; i++) {
13        double temp = crP1[i] + (alfa[i] * (crP2[i] - crP1[i]));
14        if (temp < 1) {
15            offSpr[0][i] = crP1[i];
16        } else {
17            offSpr[0][i] = temp;
18        }
19    }
20
21    for (int i = 0; i < nGen; i++) {
22        double temp = crP2[i] + (alfa[i] * (crP1[i] - crP2[i]));
23        if (temp < 1) {
24            offSpr[1][i] = crP2[i];
25        } else {
26            offSpr[1][i] = temp;
27        }
28    }
29
30    return offSpr;
31 }

```

Gambar 5.2 Source Code Proses Crossover

Penjelasan *source code* pada Gambar 5.2 mengenai proses *crossover* adalah sebagai berikut:

Baris 2-6 : pengacakan *parent* yang akan dipilih untuk proses *crossover*

Baris 13-20 : perhitungan *crossover* untuk kromosom *child 1*

Baris 22-29 : perhitungan *crossover* untuk kromosom *child 2*

5.1.3 Implementasi Proses Mutasi

Implementasi proses mutasi merupakan proses kedua pada tahap reproduksi setelah dilakukannya proses *crossover*. Proses mutasi ini mengambil acak salah satu kromosom *parent* dan memodifikasi salah satu gennya sehingga menghasilkan satu kromosom *child* baru. Proses mutasi yang digunakan yakni *random mutation*. Implementasi program proses mutasi dapat dilihat pada Gambar 5.3.

```

1 public double[] mutation() {
2     int p1 = randomParent();
3     int gen = randomGenPos();
4     double[] offspr = new double[nGen];
5     for (int i = 0; i < offspr.length; i++) {
6         offspr[i] = parent[p1][i];
7     }
8
9     double temp = offspr[gen] + randomR() * (getMaxOffsp(gen) -
10    getMinOffsp(gen));
11    if (temp < 1) {

```

```

11     return offspr;
12     } else {
13         offspr[gen] = temp;
14         return offspr;
15     }
16 }

```

Gambar 5.3 Source Code Proses Mutasi

Penjelasan *source code* pada Gambar 5.3 mengenai proses mutasi adalah sebagai berikut :

Baris 2-3 : mengacak kromosom *parent* dan gen terpilih untuk mutasi

Baris 9 : perhitungan mutasi gen untuk kromosom *child*

Baris 10-14 : pengecekan jika nilai gen berada di luar interval gen

5.1.4 Implementasi Proses Seleksi

Implementasi proses seleksi merupakan proses akhir dari algoritma genetika. Proses seleksi dilakukan untuk memilih individu-individu terbaik diantara populasinya. Proses seleksi yang digunakan yakni *elitism selection*. Implementasi program proses seleksi dapat dilihat pada Gambar 5.4.

```

1  public void elitismSelection() {
2      int n = parent.length + offspring.length;
3      double[][] allPop = new double[n][];
4      double[] allFit = new double[n];
5
6      System.arraycopy(parent, 0, allPop, 0, parent.length);
7      System.arraycopy(offspring, 0, allPop, parent.length,
8          offspring.length);
9
10     System.arraycopy(fitnesP, 0, allFit, 0, fitnesP.length);
11     System.arraycopy(fitnesC, 0, allFit, fitnesP.length,
12         fitnesC.length);
13
14     for(int i=0;i<allFit.length;i++) {
15         for (int j=i; j<allFit.length; j++) {
16             if (allFit[i] < allFit[j]) {
17                 double[] tempP = allPop[i];
18                 double tempF = allFit[i];
19                 allPop[i] = allPop[j];
20                 allFit[i] = allFit[j];
21                 allPop[j] = tempP;
22                 allFit[j] = tempF;
23             }
24         }
25     }
26
27     for (int k = 0; k < popSize; k++) {
28         parent[k] = allPop[k];
29         fitnesP[k] = allFit[k];
30     }
31 }

```

Gambar 5.4 Source Code Proses Seleksi

Penjelasan *source code* pada Gambar 5.4 mengenai proses seleksi elitism adalah sebagai berikut :

Baris 2-4 : inialisasi array untuk populasi gabungan kromosom dan *fitness*

- Baris 6-9 : penggabungan *populasi parent* dan *children* kromosom dan *fitness*
- Baris 12- 23 : proses *sorting* populasi dari *fitness* tertinggi.
- Baris 29-32 : Mengganti data *parent* dengan hasil kromosom terbaik

5.2 Implementasi Antarmuka Program

Implementasi antarmuka merupakan realisasi dari rancangan antarmuka yang telah dibuat sebelumnya. Antarmuka sistem optimasi komposisi pakan sapi perah ini terdiri dari antarmuka halaman input data dan antarmuka halaman hasil optimasi.

5.2.1 Implementasi Halaman Input Data

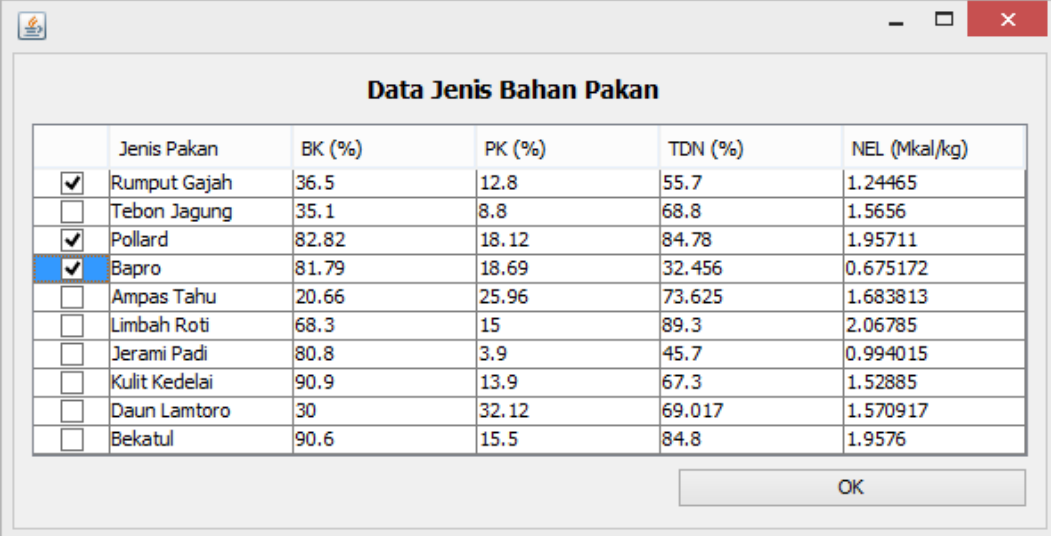
Halaman input data merupakan halaman utama dalam sistem ini. Dalam halaman ini user menginputkan semua data yang dibutuhkan mulai dari data sapi dan data pakan. Pada halaman ini pula dilakukan input data untuk parameter algoritma genetika mulai dari interval gen, nilai *popsiz*e, jumlah generasi, dan nilai *cr* dan *mr*. Hasil implementasi antarmuka halaman input data dapat dilihat pada Gambar 5.5

Bahan Pakan	BK (%)	PK (%)	TDN (%)	NEL (Mkal)	Harga
Rumput Gajah	36.5	12.8	55.7	1.24465	1250
Pollard	82.82	18.12	84.78	1.95711	4500
Bapro	81.79	18.69	32.456	0.675172	3000

Gambar 5.5 Implementasi Halaman Input Data

Dalam Gambar 5.5 user pertama menginputkan data sapi yang kemudian kebutuhannya dapat ditampilkan pada panel kebutuhan nutrisi sapi dengan menekan tombol Hitung Kebutuhan. Kemudian tombol Input Pakan digunakan untuk memilih pakan seperti pada Gambar 5.6 dimana pakan yang telah dipilih akan tampil pada panel Pakan pada Gambar 5.5. Selanjutnya user mengisi variabel

algoritma genetika pada panel Variabel Algen. Kemudian user menekan tombol Hitung Komposisi untuk memulai proses optimasi.



	Jenis Pakan	BK (%)	PK (%)	TDN (%)	NEL (Mkal/kg)
<input checked="" type="checkbox"/>	Rumput Gajah	36.5	12.8	55.7	1.24465
<input type="checkbox"/>	Tebon Jagung	35.1	8.8	68.8	1.5656
<input checked="" type="checkbox"/>	Pollard	82.82	18.12	84.78	1.95711
<input checked="" type="checkbox"/>	Bapro	81.79	18.69	32.456	0.675172
<input type="checkbox"/>	Ampas Tahu	20.66	25.96	73.625	1.683813
<input type="checkbox"/>	Limbah Roti	68.3	15	89.3	2.06785
<input type="checkbox"/>	Jerami Padi	80.8	3.9	45.7	0.994015
<input type="checkbox"/>	Kulit Kedelai	90.9	13.9	67.3	1.52885
<input type="checkbox"/>	Daun Lamtoro	30	32.12	69.017	1.570917
<input type="checkbox"/>	Bekatul	90.6	15.5	84.8	1.9576

OK

Gambar 5.6 Halaman Pilih Pakan

5.2.2 Implementasi Halaman Hasil Optimasi

Halaman hasil optimasi merupakan halaman kedua dalam sistem ini. Dimana dalam halaman ini akan menampilkan individu terbaik dari setiap generasi. Pada halaman ini juga menampilkan hasil akhir komposisi pakan, estimasi pertambahan produksi, dan juga total harga pakan. Implementasi halaman hasil optimasi dapat dilihat pada Gambar 5.6.



Gambar 5.7 Implementasi Halaman Hasil Optimasi

Dari Gambar 5.7 dapat dilihat panel Individu Terbaik Tiap Generasi akan menampilkan semua individu terbaik dari setiap generasi. Kemudian untuk hasil akhir optimasi dapat dilihat pada panel Hasil Optimasi Komposisi. Pada tabel akan muncul jenis pakan dan bobot setiap pakannya. Kemudian pada panel tersebut juga akan ditampilkan estimasi pertambahan produksi susu sapi dan juga total harga semua jenis pakan.



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan hasil pengujian terhadap variabel algoritma genetika. Pengujian ini dilakukan untuk mendapatkan variabel yang tepat untuk permasalahan yang sedang diselesaikan. Variabel algoritma genetika yang diuji meliputi *popsize*, *banyaknya generasi*, nilai *cr* dan nilai *mr*, dan *interval gen*.

6.1 Pengujian Interval Gen

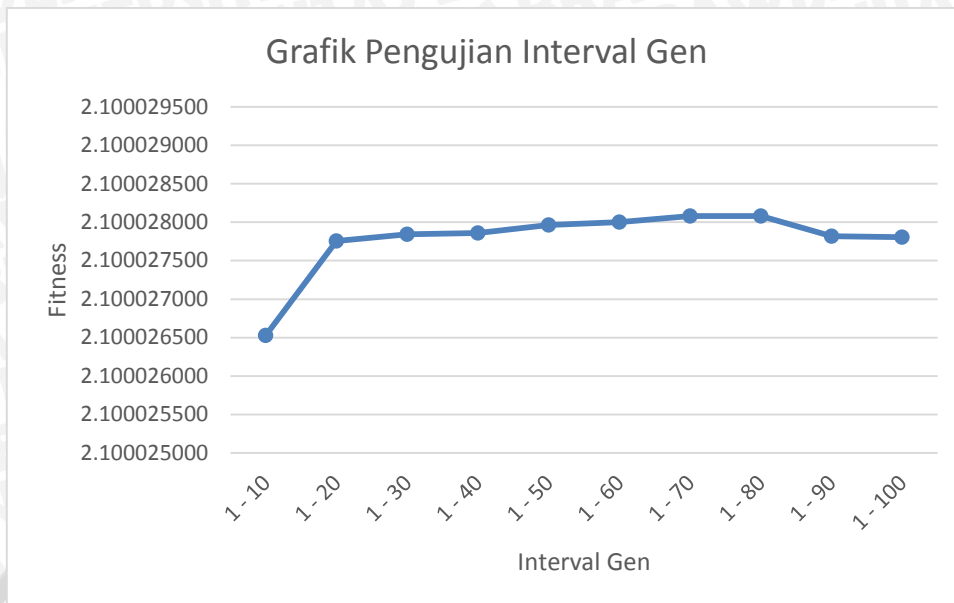
Pengujian terhadap *interval gen* dilakukan untuk mendapatkan range random gen yang optimal terhadap permasalahan optimasi komposisi pakan sapi. Data objek yang digunakan dalam pengujian ini yakni sapi dengan bobot 400 kg, produksi susu 11 liter per hari per ekor, dan kandungan lemak dalam susu sebesar 4%. Data pakan yang digunakan yakni rumput gajah, tebon jagung, dan bekatul. Sedangkan parameter algoritma genetika yang digunakan adalah nilai *popsize* 50, *banyak generasi* 100, nilai *cr* 0.4, dan nilai *mr* 0.2.

Interval gen yang diuji dimulai dari range [1:10] hingga [1:100] dengan interval 10. Nilai *interval gen* yang terbaik akan diambil dari rata-rata *fitness* terbaik dari 10 percobaan pengujian yang dilakukan. Setiap percobaan akan dibangkitkan kromosom baru secara random. Hasil pengujian terhadap *interval gen* dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Interval Gen

Interval Gen	Fitness Pada Percobaan ke - <i>i</i>					Rata-rata
	1	2	3	...	10	
1 – 10	2.1000268	2.1000267	2.1000263		2.1000265	2.100026529
1 – 20	2.1000272	2.1000282	2.1000278		2.1000281	2.100027756
1 – 30	2.1000277	2.1000269	2.1000281		2.1000278	2.100027842
1 – 40	2.1000274	2.1000283	2.1000276		2.1000282	2.100027861
1 – 50	2.1000273	2.1000281	2.1000284		2.1000280	2.100027963
1 – 60	2.1000285	2.1000271	2.1000282		2.1000284	2.100028000
1 – 70	2.1000280	2.1000272	2.1000281		2.1000278	2.100028080
1 – 80	2.1000274	2.1000278	2.1000282		2.1000284	2.100028081
1 – 90	2.1000271	2.1000273	2.1000280		2.1000283	2.100027817
1 – 100	2.1000282	2.1000280	2.1000273		2.1000273	2.100027804

Dari pengujian pada Tabel 6.1 didapatkan grafik yang menggambarkan jelas hasil pengujian terhadap *interval gen*. Grafik hasil pengujian *interval gen* dapat dilihat pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Interval Gen

Dari grafik hasil pengujian pada Gambar 6.1 dapat dilihat bahwa terjadi kenaikan yang cukup signifikan dari interval [1:10] menuju [1:20]. Selanjutnya grafik terus menunjukkan kestabilan nilai *fitness* hingga interval [1:80] walaupun terjadi penurunan nilai *fitness* yang sangat kecil hingga interval [1:100]. Dari hal tersebut bisa disimpulkan bahwa *interval gen* yang kecil akan membentuk kromosom dengan bobot pakan yang kecil pula. Sedangkan bobot yang kecil cenderung tidak bisa memenuhi kebutuhan nutrisi sapi perah. Sebaliknya nilai *interval gen* yang besar akan memberikan range pencarian gen yang besar untuk memenuhi kebutuhan nutrisi sapi perah. Dari grafik hasil pengujian dapat dilihat bahwa *interval gen* yang mendekati optimal yakni [1:80] dengan nilai *fitness* tertinggi 2.100028081.

6.2 Pengujian Ukuran Populasi

Pengujian terhadap *ukuran populasi* dilakukan untuk mendapatkan ukuran populasi yang optimal terhadap permasalahan optimasi komposisi pakan sapi. Data objek yang digunakan dalam pengujian ini yakni sapi dengan bobot 400 kg, produksi 11 liter per hari per ekor, dan kandungan lemak dalam susu sebesar 4%. Data pakan yang digunakan yakni rumput gajah, tebon jagung, dan bekatul. Sedangkan parameter algoritma genetika yang digunakan yakni *jumlah generasi* 100, nilai *cr* 0.4, nilai *mr* 0.2, dan *interval gen* [1:80].

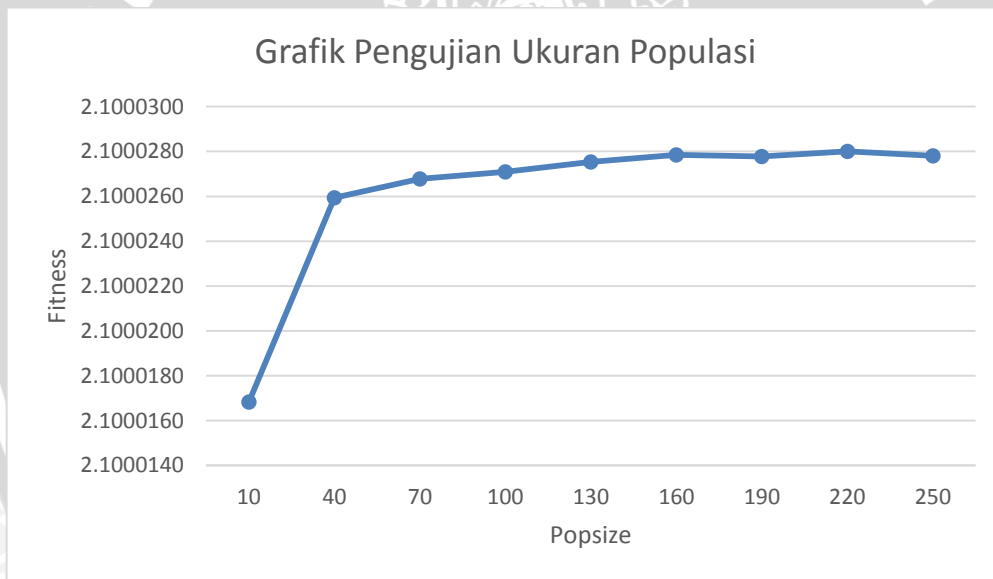
Ukuran populasi juga menentukan jumlah *child* yang terbentuk. *Ukuran populasi* yang diuji dimulai dari 10 hingga 250 dengan interval 30. Nilai *ukuran populasi* yang terbaik akan diambil dari rata-rata *fitness* terbaik dari 10 percobaan pengujian yang dilakukan. Setiap percobaan akan dibangkitkan individu baru secara *random*. Pengujian terhadap *ukuran populasi* ini dilakukan karena algoritma genetika merupakan algoritma berbasis populasi. Maka semakin besar *ukuran populasi* maka ruang pencarian terhadap solusi juga akan semakin luas,

namun tidak menjamin solusi yang dihasilkan semakin optimal (Mahmudy, 2013). Hasil pengujian *ukuran populasi* bisa dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Ukuran Populasi

Popsize	Fitness Pada Percobaan ke - <i>i</i>					Rata-rata
	1	2	3	...	10	
10	2.100016	2.100020	2.100016		2.100026	2.1000168
40	2.100025	2.100027	2.100028		2.100025	2.1000259
70	2.100025	2.100026	2.100028		2.100027	2.1000268
100	2.100028	2.100027	2.100027		2.100027	2.1000271
130	2.100027	2.100027	2.100027		2.100027	2.1000275
160	2.100028	2.100027	2.100027		2.100028	2.1000278
190	2.100028	2.100027	2.100027		2.100028	2.1000278
220	2.100028	2.100028	2.100028		2.100028	2.1000280
250	2.100027	2.100028	2.100027		2.100028	2.1000278

Dari pengujian pada Tabel 6.2 didapatkan grafik yang menggambarkan dengan jelas hasil pngujian terhadap *ukuran populasi*. Grafik hasil pengujian *ukuran populasi* dapat dilihat pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Pengujian Ukuran Populasi

Grafik hasil uji *ukuran populasi* pada Gambar 6.2 menyatakan bahwa *ukuran populasi* 220 memiliki *fitness* terbesar diantara nilai *ukuran populasi* lainnya yakni dengan *fitness* 2.1000280. Terjadi peningkatan yang cukup signifikan pada *ukuran populasi* 40, namun pada nilai 40 hingga 250 sudah menunjukkan kestabilan nilai *fitness* walaupun masih mengalami peningkatan dan penurunan yang cukup kecil. Nilai *ukuran populasi* yang besar akan memang akan meningkatkan eksplorasi algoritma genetika dalam mencari permasalahan, namun nilai *ukuran populasi* yang besar juga akan memperlambat waktu komputasi (Mahmudy, 2013). Dari Gambar 6.2 juga dapat disimpulkan bahwa *ukuran populasi* yang kecil akan membatasi algoritma genetika dalam mengeksplorasi solusi.

6.3 Pengujian Jumlah Generasi

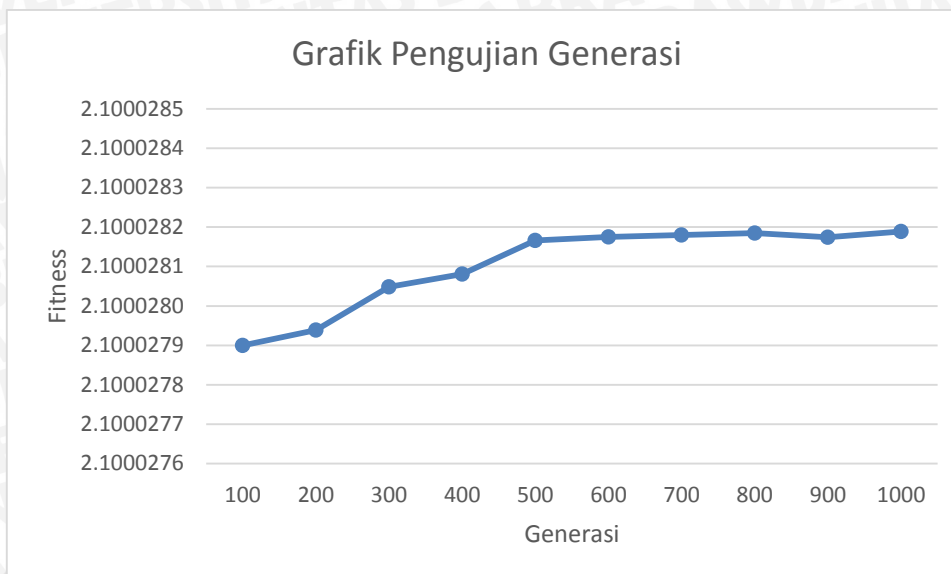
Pengujian terhadap *jumlah generasi* dilakukan untuk mendapatkan banyaknya generasi yang optimal terhadap permasalahan optimasi komposisi pakan sapi. Data objek yang digunakan dalam pengujian ini yakni sapi dengan bobot 400 kg, produksi 11 liter per hari per ekor, dan kandungan lemak dalam susu sebesar 4%. Data pakan yang digunakan yakni rumput gajah, tebon jagung, dan bekatul. Sedangkan parameter algoritma yang digunakan yakni nilai cr 0.4, nilai mr 0.2, *interval gen* [1:18], dan *ukuran populasi* menggunakan hasil pengujian ukuran populasi sebelumnya yakni 220.

Jumlah generasi yang diuji dimulai dari 100 hingga 1000 dengan interval 100. *Jumlah generasi* yang terbaik akan diambil dari rata-rata *fitness* terbaik dari 10 percobaan pengujian yang dilakukan. Setiap percobaan akan dibangkitkan kromosom baru secara random. Hasil pengujian terhadap *jumlah generasi* dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian Banyaknya Generasi

Generasi	Fitness Pada Percobaan ke - <i>i</i>					Rata-rata
	1	2	3	...	10	
100	2.1000283	2.1000268	2.1000285		2.1000272	2.10002790
200	2.1000279	2.1000270	2.1000282		2.1000282	2.10002794
300	2.1000276	2.1000279	2.1000285		2.1000283	2.10002805
400	2.1000281	2.1000283	2.1000279		2.1000277	2.10002808
500	2.1000279	2.1000282	2.1000281		2.1000280	2.10002817
600	2.1000285	2.1000278	2.1000282		2.1000283	2.10002817
700	2.1000281	2.1000282	2.1000284		2.1000272	2.10002818
800	2.1000280	2.1000280	2.1000280		2.1000284	2.10002818
900	2.1000285	2.1000285	2.1000278		2.1000282	2.10002817
1000	2.1000283	2.1000283	2.1000283		2.1000278	2.10002819

Dari hasil pengujian pada Tabel 6.3 didapatkan grafik yang menggambarkan jelas hasil pengujian terhadap *jumlah generasi*. Grafik hasil pengujian *jumlah generasi* dapat dilihat pada Gambar 6.3.



Gambar 6.3 Grafik Pengujian Jumlah Generasi

Dari grafik hasil pengujian *jumlah generasi* pada Gambar 6.3 menunjukkan terjadi peningkatan yang cukup signifikan. Pada generasi 100 hingga 500 terjadi kenaikan karena seiring bertambahnya nilai generasi maka akan memperluas ruang pencarian solusi. Kemudian nilai *fitness* cenderung stabil hingga generasi 1000 walaupun terdapat sedikit kenaikan dan penurunan. Dari grafik hasil uji *jumlah generasi* menunjukkan bahwa nilai generasi 1000 adalah yang mendekati optimal dengan nilai *fitness* tertinggi yakni 2.10002819.

6.4 Pengujian Nilai *Cr* dan *Mr*

Pengujian terhadap nilai *cr* dan *mr* dilakukan untuk mendapatkan kombinasi nilai *cr* dan *mr* yang optimal terhadap permasalahan optimasi komposisi pakan sapi. Kombinasi nilai *cr* dan *mr* akan menentukan jumlah *child* yang terbentuk dari sebuah populasi *parent*. Data objek yang digunakan dalam pengujian ini yakni sapi dengan bobot 400 kg, produksi susu 11 liter per hari per ekor, dan kandungan lemak dalam susu sebesar 4%. Data pakan yang digunakan yakni rumput gajah, tebon jagung, dan bekatul. Sedangkan parameter algoritma genetika yang digunakan yakni nilai *popsiz* 220, banyaknya *generasi* 1000, dan *interval gen* [1:80].

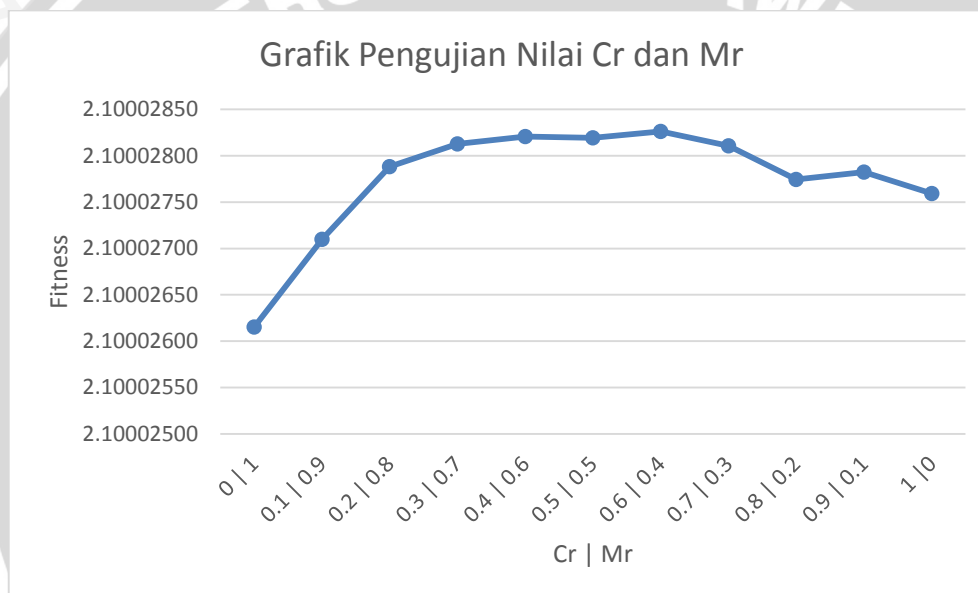
Nilai *cr* dan *mr* yang diuji dimulai dari 0 hingga 1. Nilai kombinasi *cr* dan *mr* yang terbaik akan diambil dari rata-rata *fitness* terbaik dari 10 percobaan pengujian yang dilakukan. Setiap percobaan akan dibangkitkan kromosom baru secara random. Hasil pengujian terhadap kombinasi nilai *cr* dan *mr* dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Pengujian Kombinasi Nilai *Cr* dan *Mr*

<i>cr</i> <i>mr</i>	<i>Fitness</i> Pada Percobaan ke - <i>i</i>					Rata-rata
	1	2	3	...	10	
0 1	2.1000272	2.1000278	2.1000255		2.1000249	2.10002615
0.1 0.9	2.1000285	2.1000276	2.1000270		2.1000262	2.10002710

cr mr	Fitness Pada Percobaan ke - i					Rata-rata
	1	2	3	...	10	
0.2 0.8	2.1000273	2.1000283	2.1000283		2.1000282	2.10002788
0.3 0.7	2.1000285	2.1000284	2.1000284		2.1000276	2.10002813
0.4 0.6	2.1000283	2.1000285	2.1000274		2.1000285	2.10002821
0.5 0.5	2.1000282	2.1000285	2.1000284		2.1000285	2.10002819
0.6 0.4	2.1000282	2.1000283	2.1000281		2.1000285	2.10002826
0.7 0.3	2.1000285	2.1000282	2.1000279		2.1000282	2.10002811
0.8 0.2	2.1000267	2.1000275	2.1000281		2.1000282	2.10002774
0.9 0.1	2.1000279	2.1000285	2.1000275		2.1000273	2.10002782
1 0	2.1000280	2.1000276	2.1000274		2.1000277	2.10002759

Dari pengujian pada Tabel 6.4 didapatkan grafik yang menggambarkan jelas hasil pengujian terhadap kombinasi *cr* dan *mr*. Grafik hasil pengujian kombinasi nilai *cr* dan *mr* dapat dilihat pada Gambar 6.4.



Gambar 6.4 Grafik Hasil Pengujian Kombinasi Nilai *Cr* dan *Mr*

Dari grafik hasil pengujian 6.4 dapat dilihat bahwa terjadi peningkatan pada kombinasi nilai *cr* dan *mr* 0,1 hingga 0,3,0,7 kemudian nilai fitness mengalami kestabilan walaupun terjadi sedikit kenaikan dan penurunan. Kemudian terjadi penurunan nilai fitness pada kombinasi nilai *cr* 0.7 dan *mr* 0.3. Dari penurunan tersebut dapat disimpulkan bahwa nilai *cr* dan *mr* sangat berpengaruh terhadap pencarian solusi. Sedangkan untuk mendapatkan solusi yang mendekati optimal keseimbangan nilai *cr* dan *mr* sangat penting untuk eksplorasi dan eksploitasi solusi. Nilai *crossover rate* yang rendah dan mutasi yang tinggi memang tidak akan mempengaruhi eksplorasi solusi, namun hal tersebut akan menyebabkan algoritma tidak bisa ‘belajar’ dari generasi sebelumnya. Akibatnya algoritma akan kesulitan dalam mengeksplorasi ruang pencarian solusi (Mahmudy, 2013).

Ketidakseimbangan juga akan terjadi jika nilai *crossover rate* yang tinggi dan *mutation rate* yang rendah. Nilai *crossover rate* yang tinggi akan menyebabkan *offspring* yang dihasilkan cenderung memiliki kemiripan kromosom yang tinggi

dengan induknya sehingga algoritma kesulitan dalam melakukan eksplorasi dalam ruang pencarian solusi (Mahmudy, 2013). Dari grafik hasil pengujian terlihat bahwa nilai kombinasi yang paling optimal yakni cr 0.6 dan mr 0.4 yang ditunjukkan dengan nilai *fitness* tertinggi yakni 2.10002826.

6.5 Evaluasi Hasil Optimasi

Dari pengujian terhadap parameter algoritma genetika yang telah dilakukan, didapat parameter yang tepat dengan *interval gen* [1:80], *popsiz*e 80, banyak *generasi* 1000, nilai cr 0.6, dan nilai mr 0.4.

Sistem tersebut kemudian akan dievaluasi dengan membandingkan hasil optimasi dari sistem dengan data studi kasus yang didapatkan dari peternak di kota Batu. Evaluasi ini dilakukan dengan menghitung komposisi bahan pakan yang sama antara komposisi yang diberikan oleh peternak dan komposisi hasil optimasi dari sistem. Data pemberian pakan sapi oleh peternak dapat dilihat pada Tabel 6.5.

Tabel 6.5 Data Hasil Observasi Peternak

No.	Keadaan sapi	Pakan	Jumlah	Harga	Produksi susu
1.	BB : 400 kg Produksi : 8 L Lemak : 4%	Rumput gajah	30 kg	Rp 1250 x 30 = Rp 37500	8 L
		Bapro	1.5 kg	Rp 3000 x 1.5 = Rp 4500	
		Pollard	1.5 kg	Rp 4500 x 1.5 = Rp 6750	
Jumlah			33 kg	Rp 48750	
2.	BB : 400 kg Produksi : 9 L Lemak : 4.2%	Rumput gajah	25 kg	Rp 1250 x 25 = Rp 31250	9 L
		Pollard	4 kg	Rp 4500 x 4 = Rp 18000	
		Jumlah			
3.	BB : 380 kg Produksi : 6 L Lemak : 4.2%	Rumput gajah	25 kg	Rp 1250 x 25 = Rp 31250	6 L
		Pollard	2,5 kg	Rp 3000 x 2.5 = Rp 7500	
		Jumlah			
4.	BB : 400 kg Produksi : 8 L Lemak : 4.2%	Rumput gajah	25 kg	Rp 1250 x 25 = Rp 31250	8 L
		Pollard	3 kg	Rp 4500 x 3 = Rp 13500	
		Jumlah			
5.	BB : 400 kg Produksi : 8 L Lemak : 4.2%	Rumput gajah	30 kg	Rp 1250 x 30 = Rp 37500	8 L
		Pollard	3 kg	Rp 4500 x 3 = Rp 13500	
		Jumlah			

Pada Tabel 6.5 data keadaan sapi dan pakan yang diberikan diambil dari data hasil observasi dari peternak dan KUD yang meliputi berat badan, produksi, dan kandungan lemak. Kemudian pakan yang diberikan dihitung total harganya berdasarkan harga tiap-tiap bahan pakan. Dengan objek sapi dan bahan pakan yang sama akan dilakukan perhitungan komposisi pakan dengan menggunakan algoritma genetika. Hasil optimasi komposisi pakan oleh sistem dapat dilihat pada Tabel 6.6.

Tabel 6.6 Hasil Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Genetika

No.	Keadaan sapi	Pakan	Jumlah	Harga	Pertambahan Produksi
1.	BB : 400 kg Produksi : 8 L Lemak : 4%	Rumput gajah	14.5 kg	Rp 1250 x 14.5 = Rp 18125	0.8 L
		Bapro	5 kg	Rp 3000 x 5 = Rp 15000	
		Pollard	1 kg	Rp 4500 x 1 = Rp 4500	
Jumlah			20.5 kg	Rp 37625	
2.	BB : 400 kg Produksi : 9 L Lemak : 4.2%	Rumput gajah	33 kg	Rp 1250 x 33 = Rp 41250	0.9 L
		Pollard	1 kg	Rp 4500 x 1 = Rp 4500	
		Jumlah			
3.	BB : 380 kg Produksi : 6 L Lemak : 4.2%	Rumput gajah	23.7 kg	Rp 1250 x 23.7 = Rp 29625	0.6 L
		Pollard	1 kg	Rp 4500 x 1 = Rp 4500	
		Jumlah			
4.	BB : 400 kg Produksi : 8 L Lemak : 4.2%	Rumput gajah	30 kg	Rp 1250 x 30 = Rp 37500	0.8 L
		Pollard	1 kg	Rp 4500 x 1 = Rp 4500	
		Jumlah			
5.	BB : 400 kg Produksi : 8 L Lemak : 4.2%	Rumput gajah	30.5 kg	Rp 1250 x 30 = Rp 38125	0.8 L
		Pollard	1 kg	Rp 4500 x 1 = Rp 4500	
		Jumlah			

Dari Tabel 6.6 diperoleh komposisi pakan sapi perah hasil optimasi dari sistem dengan menggunakan algoritma genetika. Selanjutnya dilakukan perbandingan harga dan pertambahan produksi dari komposisi milik peternak dan hasil sistem. Data perbandingan tersebut dapat dilihat pada Tabel 6.7.

Tabel 6.7 Perbandingan Komposisi Peternak dan Sistem

No.	Hasil Observasi		Hasil Sistem		Selisih	
	Biaya Pakan	Pertambahan produksi susu	Biaya Pakan	Pertambahan produksi susu	Biaya Pakan	Pertambahan produksi susu
1.	Rp 48750	0 L	Rp 37625	0.8 L	Rp 11125	0.8 L
2.	Rp 49250	0 L	Rp 45750	0.9 L	Rp 3500	0.9 L
3.	Rp 37750	0 L	Rp 34125	0.6 L	Rp 3625	0.6 L
4.	Rp 44750	0 L	Rp 42000	0.8 L	Rp 2750	0.8 L
5.	Rp 51000	0 L	Rp 42625	0.8 L	Rp 8375	0.8 L
Jumlah					Rp 23975	3.9 L
Rata-Rata					Rp 5875	0.78 L (10% dari produksi rata-rata)

Dari hasil perbandingan pada Tabel 6.7 dapat disimpulkan bahwa sistem optimasi komposisi pakan sapi perah menggunakan algoritma genetika dapat menghemat sebesar Rp 5875 untuk tiap kali pemberian makan dengan estimasi pertambahan produksi yakni 10% dari produksi sebenarnya.

BAB 7 PENUTUP

Bab ini menjelaskan kesimpulan yang didapat dari penelitian yang telah dilakukan dan saran yang untuk menyempurnakan penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan berdasarkan hasil pengujian serta analisis terhadap sistem optimasi komposisi pakan sapi perah menggunakan algoritma genetika dapat disimpulkan beberapa hal yakni sebagai berikut :

1. Algoritma genetika dapat digunakan untuk menyelesaikan permasalahan optimasi komposisi pakan. Dengan representasi kromosom real code, hasil komposisi pakan yang dihasilkan mampu menekan biaya pakan dibandingkan biaya yang biasa dikeluarkan oleh peternak. Dari representasi itu pula didapatkan estimasi pertambahan produksi susu bagi sapi.
2. Berdasarkan pengujian yang telah dilakukan terhadap parameter algoritma genetika, penentuan parameter yang tepat sangat mempengaruhi solusi yang dihasilkan. Dari hasil pengujian tersebut didapatkan parameter algoritma genetika yakni interval gen [1:80], ukuran populasi 220, banyak generasi 1000, nilai cr 0.6, dan nilai mr 0.4.
3. Dalam mengukur kualitas hasil optimasi menggunakan sistem yang telah dibangun, dilakukan global analisis terkait solusi yang dihasilkan sistem dengan kondisi sebenarnya di lapangan, dalam hal ini peternak di kota Batu. Hasilnya menunjukkan bahwa hasil optimasi sistem dapat mengemat Rp 5875 untuk setiap kali pemberian pakan pada seekor sapi dengan estimasi pertambahan produksi 10% dari produksi sebenarnya.

7.2 Saran

Saran yang didapatkan untuk pengembangan terhadap penelitian ini terdapat beberapa hal :

1. Karena dalam sistem belum ada perhitungan perbandingan antaranya banyaknya pakan hijauan dan konsentrat, maka untuk pengembangannya dapat dimasukkan perbandingan antara pakan hijauan dan konsentrat dalam perhitungan fitnessnya sehingga kromosom pakan yang dihasilkan memiliki perbandingan yang baik antara hijauan dan konsentrat.
2. Karena terdapat beberapa acuan dalam menghitung kebutuhan sapi perah dan setiap metode menghitung dari jenis nutrisi yang berbeda, maka pengembangan dapat dilakukan dengan melakukan perhitungan dari acuan yang lain. Dan karena bahan pakan yang dioptimasi hanya terdapat 10 bahan pakan, maka dalam pengembangan selanjutnya diharapkan ditambahkan bahan pakan lain sehingga hasilnya lebih bervariasi.

3. Sistem ini hanya dapat menghitung kebutuhan dan memberikan rekomendasi komposisi pakan untuk tiap ekor saja. Sehingga jika diterapkan pada peternakan sapi perah yang berskala besar akan sangat memakan waktu bagi peternak untuk menghitung komposisi pakan setiap ekor sapi. Sehingga dalam pengembangan selanjutnya sistem dapat dibuat agar bisa menghitung komposisi pakan untuk beberapa ekor sapi sekaligus.



DAFTAR PUSTAKA

- Council, N. R., 1978. *Nutrient Requirement of Dairy Cattle Fifth Revised Edition*. 5th penyunt. Washington, D.C.: National Academy of Sciences.
- Council, N. R., 2001. *Nutrient Requirements of Dairy Cattle Seventh Revised Edition*. 7th penyunt. Washington, D.C.: National Academy of Sciences.
- Fakhiroh, D., Mahmudy, W. F. & Indriati, 2015. Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Genetika. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*.
- Fanjiang, Y.-Y. & Syu, Y., 2014. Semantic-based Automatic Service Composition With Functional and Non-functional Requirements in Design time: A Genetic Algorithm Approach. *Information and Software Technology*, Volume 56, pp. 352-373.
- Farid, M. & Sukesi, H., 2011. Pengembangan Susu Segar Dalam Negeri Untuk Pemenuhan Kebutuhan Susu Nasional. *Buletin Ilmiah Litbang Perdagangan*, 5(2), pp. 196-221.
- Gen, M. & Cheng, R., 2000. *Genetic Algorithms and Engineering Optimization*. Toronto: A Wiley-Interscience Publication.
- Kaya, Y., Uyar, M. & Tekin, R., 2012. A Novel Crossover Operator for Genetic Algorithms: Ring Crossover. Volume 1, pp. 1-4.
- Li, S., Yan, Z., Jian, L. & Xu, J., 2015. *Study on Auto Parts Suppliers Composition Selection Based on Adaptive Genetic Algorithm*. Leicester, IEEE.
- Magalhaes-Mendes, J., 2013. A Comparative Study of Crossover Operatrs for Genetic Algorithms to Solve the Job Shop Scheduling Problem. *WSEAS Transactions on Computers*, 12(4), pp. 164-173.
- Mahmudy, W. F., 2013. *Algoritma Evolusi*. Malang: s.n.
- Mahmudy, W. F., Marian, R. M. & Luong, L. H. S., 2014. Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System. *ECTI Transactions On Computer And Information Technology*, 8(1), pp. 80-92.
- Mansyur, et al., 2008. *Kemandirian Pakan Dalam Pengembangan Sapi Perah (Kasus KSU Tandangsari Sumedang)*. Yogyakarta, Fakultas Peternakan UGM.
- Marginingtyas, E., Mahmudy, W. F. & Indriati, 2015. Penentuan Komposisi Pakan Ternak Untuk Memenuhi Kebutuhan Nutrisi Ayam Petelur Dengan Biaya Minimum Menggunakan Algoritma Genetika. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(12).
- Miller, W. J., 1979. *Dairy Cattle Feeding and Nutrition*. London: Academic Press.

- Mitchell, M., 1999. *An Introduction To Genetic Algorithm*. 5th penyunt. Massachusetts: The MIT Press.
- Muliantara, A., 2012. Penentuan Komposisi Bahan Pakan Ikan Lele Yang Optimal Dengan Menggunakan Metode IWO-Subtractive Clustering. *Jurnal Ilmu Komputer*, 5(2), pp. 23-28.
- Novianti, J., 2014. *Respon Fisiologis Sapi Perah FH Laktasi Yang Diberikan Pakan Rumput Gajah (Pennisetum Purpureum) Dengan Ukuran Potongan Yang Berbeda*, Bogor: Sekolah Pascasarjana Institut Pertanian Bogor.
- Rothlauf, F., 2006. *Representations for Genetic and Evolutionary Algorithms*. 2nd penyunt. Berlin: Springer.
- Sivanandam, S. N. & Deepa, S. N., 2008. *Introduction to Genetic Algorithms*. New York: Springer-Verlag.
- Statistik, B. P., 2015. *Produksi Susu Segar Menurut Provinsi 2009-2015*. [Online] Available at: <http://www.bps.go.id/linkTableDinamis/view/id/1083> [Diakses 25 January 2015].
- Wahid, N. & Mahmudy, W. F., 2015. Optimasi Komposisi Makanan Untuk Penderita Koesterol Menggunakan Algoritma Gentika. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(15).
- Wang, L., Shen, J., Luo, J. & Dong, F., 2013. *An Improved Genetic Algorithm for Cost-Effective Data-Intensive Service Composition*. Beijing, IEEE.
- Wijaya, R. B., 2015. *Manajemen Pemberian Pakan Sapi Perah Periode Laktasi Di Koperasi Unit Desa (KUD) Batu*, Malang: Fakultas Peternakan Universitas Brawijaya.
- Yudhoyono, S. B., 2008. *Peraturan Presiden Republik Indonesia*. Indonesia, Paten No. 28.
- Zhao, Z., Wang, S. & Hong, X., 2015. *A Web Service Composition Method Based on Merging Genetic Algorithm and Ant Colony Algorithm*. Liverpool, IEEE.

LAMPIRAN A HASIL WAWANCARA NARASUMBER

Pada lampiran ini dituliskan hasil wawancara dengan Dr. Ir. Irfan H. Djunaidi, M.Sc selaku dosen Fakultas Peternakan bagian Nutrisi dan Makanan Ternak yang membahas tentang sapi perah, komposisi pakan sapi perah, dan produksi susu.

Pertanyaan :

Bagaimanakah cara yang baik dalam menentukan kebutuhan nutrisi sapi perah?

Jawaban :

Untuk menentukan kebutuhan sapi perah yang pertama berdasarkan bobot sapi dan yang kedua jumlah produksi. Bobot badan ini nanti digunakan untuk menentukan kebutuhan minimal yakni kebutuhan hidup pokok, karena tanpa berproduksi sapi perah tetap membutuhkan nutrisi. Semakin besar bobotnya maka kebutuhannya nutrisinya semakin banyak. Kebutuhan ini nantinya akan digunakan untuk melangsungkan hidup. Di atas itu juga ada kebutuhan nutrisi untuk per liter susu yakni disebut kebutuhan produksi. Kebutuhan produksi tentu saja tergantung pada jumlah produksi. Kemudian yang kedua, karena kualitas susu salah satunya ditentukan oleh kadar lemak, sehingga setiap perbedaan kadar lemak juga membutuhkan nutrisi yang berbeda pula. Untuk perhitungan dalam menentukan komposisi pakan sapi perah bisa mengacu pada laporan KKN Riyan Budi Wijaya terkait manajemen pemberian pakan sapi perah di KUD Batu. Kemudian karena kandungan nutrisi bahan pakan di setiap daerah berbeda maka acuan bahan pakan beserta nutrisinya bisa dilihat di NRC 2001.

Pertanyaan :

Apakah terdapat patokan produksi susu yang dihasilkan dari sapi dengan berat badan tertentu?

Jawaban :

Kemampuan sapi berproduksi ditentukan oleh genetiknya. Jika ingin memproduksi susu banyak maka sapi harus mempunyai potensi genetik yang tinggi. Jadi untuk meningkatkan produksi susu sangatlah sulit bahkan mungkin hanya 10% saja. Sedangkan pakan dan lingkungan hanya mendukung potensi genetik sapi yang diartikan memaksimalkan potensi produksi. Sapi dengan bobot 450kg mampu menghasilkan 15 liter susu, namun kenyataannya hanya menghasilkan 8 liter susu, maka produksi tersebut dapat dimaksimalkan hingga 15 liter.

Pertanyaan :

Apakah dapat dihitung komposisi minimal pakan yang dibutuhkan seekor sapi perah dengan berat badan, jumlah produksi, dan kandungan lemak tertentu?

Jawaban :

Tidak ada penentuan komposisi pakan minimal untuk setiap sapi. Kembali lagi untuk menentukan kebutuhan pakan harus dihitung terlebih dahulu kebutuhan nutrisinya. Dari data berat badan, produksi, dan kandungan lemak susu sapi dapat

ditentukan berapa kebutuhan tiap nutrisinya. Kemudian kebutuhan nutrisi tersebut ditotal dan barulah dari total nutrisi tersebut dapat dihitung kebutuhan pakannya berapa. Jenis pakan sapi perah harus mengandung hijauan dan konsentrat.



LAMPIRAN B HASIL OBSERVASI DATA SAPI DI KUD BATU DAN DESA DRESEL, KECAMATAN BATU, KOTA BATU

Pada lampiran ini dituliskan data rutinitas pemberian pakan sapi dan produksi sapi yang diambil dari peternak di Kota batu beserta kandungan lemak susu sapi yang diperoleh dari KUD Batu.

Maskur Afandi (664)

No. Sapi	Hari, tanggal	Pakan				Produksi susu	Kadar Lemak
		Hijauan	Jumlah	Konsentrat	Jumlah		
1	Selasa, 15-12-2015	P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	4%
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	
3		P. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	5 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	3 L	
4		P. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	4 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	3 L	
1	Rabu, 16-12-2015	P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	
3		P. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	5 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	3 L	
4		P. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	4 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	2.5 kg	3 L	
1	Kamis, 16-12-2015	P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	
3		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	3 L	
4		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	4 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	3 L	
1	Jumat, 17-12-2015	P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	
2		P. Rumput gajah	30 kg	Bapro/Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Bapro/Pollard	3 kg	5 L	

Dosan (675)

No. Sapi	Hari, tanggal	Pakan				Produksi susu	Kadar Lemak
		Hijauan	Jumlah	Konsentrat	Jumlah		
1	Selasa, 15-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	9 L	4,2%
		S. Rumput gajah	25 kg	Pollard	4 kg	5 L	
2		P. Rumput gajah	25 kg	Pollard	4 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L	
1	Rabu, 16-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L	
2		P. Rumput gajah	25 kg	Pollard	4 kg	9 L	

1	Kamis, 17-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
S. Rumput gajah		25 kg	Pollard	4 kg	5 L	
P. Rumput gajah		25 kg	Pollard	4 kg	8 L	
2	Jumat, 18-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
S. Rumput gajah		25 kg	Pollard	4 kg	5 L	
P. Rumput gajah		25 kg	Pollard	4 kg	8 L	
1	Sabtu, 19-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
S. Rumput gajah		25 kg	Pollard	4 kg	5 L	
P. Rumput gajah		25 kg	Pollard	4 kg	8 L	
2	Minggu, 20-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
S. Rumput gajah		25 kg	Pollard	4 kg	5 L	
P. Rumput gajah		25 kg	Pollard	4 kg	8 L	
1	Minggu, 20-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
S. Rumput gajah		25 kg	Pollard	4 kg	5 L	
P. Rumput gajah		25 kg	Pollard	4 kg	8 L	

Kartubi (680)

No. Sapi	Hari, tanggal	Pakan				Produksi susu	Kadar Lemak
		Hijauan	Jumlah	Konsentrat	Jumlah		
1	Selasa, 15-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	4,2%
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Rabu, 16-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Kamis, 17-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Jumat, 18-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Sabtu, 19-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Minggu, 20-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	
1	Senin, 21-12-2015	P. Rumput gajah	25 kg	Pollard	2,5 kg	6 L	
		S. Rumput gajah	25 kg	Pollard	2,5 kg	4 L	

Siono (685)

No. Sapi	Hari, tanggal	Pakan				Produksi susu	Kadar Lemak
		Hijauan	Jumlah	Konsentrat	Jumlah		
1	Selasa, 15-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	4,2%
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Rabu, 16-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Kamis, 17-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	9 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Jumat, 18-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	4 L	
1	Sabtu, 19-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Minggu,	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	

	20-12-2015	S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Senin, 21-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	

Eko (2584)

No. Sapi	Hari, tanggal	Pakan				Produksi susu	Kadar Lemak
		Hijauan	Jumlah	Konsentrat	Jumlah		
1	Selasa, 15-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	4,2%
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Rabu, 16-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Kamis, 17-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Jumat, 18-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Sabtu, 19-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Minggu, 20-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
1	Senin, 21-12-2015	P. Rumput gajah	30 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	
2		P. Rumput gajah	30 kg	Pollard	3 kg	6 L	
		S. Rumput gajah	30 kg	Pollard	3 kg	4 L	



LAMPIRAN C HASIL PENGUJIAN ALGORITMA GENETIKA

Hasil Pengujian Interval Gen

Interval	Fitness Percobaan ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
1 - 10	2.1000268	2.1000267	2.1000263	2.1000267	2.1000271	2.1000266	2.1000263	2.1000256	2.1000266	2.1000265	2.100026529
1 - 20	2.1000272	2.1000282	2.1000278	2.1000283	2.1000278	2.1000276	2.1000277	2.1000273	2.1000277	2.1000281	2.100027756
1 - 30	2.1000277	2.1000269	2.1000281	2.1000281	2.1000282	2.1000274	2.1000281	2.1000282	2.1000279	2.1000278	2.100027842
1 - 40	2.1000274	2.1000283	2.1000276	2.1000274	2.1000279	2.1000281	2.1000281	2.1000277	2.1000280	2.1000282	2.100027861
1 - 50	2.1000273	2.1000281	2.1000284	2.1000272	2.1000280	2.1000281	2.1000281	2.1000282	2.1000283	2.1000280	2.100027963
1 - 60	2.1000285	2.1000271	2.1000282	2.1000282	2.1000283	2.1000282	2.1000273	2.1000281	2.1000278	2.1000284	2.100028000
1 - 70	2.1000280	2.1000272	2.1000281	2.1000285	2.1000282	2.1000282	2.1000283	2.1000281	2.1000285	2.1000278	2.100028080
1 - 80	2.1000274	2.1000278	2.1000282	2.1000281	2.1000281	2.1000283	2.1000281	2.1000283	2.1000281	2.1000284	2.100028081
1 - 90	2.1000271	2.1000273	2.1000280	2.1000278	2.1000277	2.1000282	2.1000272	2.1000284	2.1000282	2.1000283	2.100027817
1 - 100	2.1000282	2.1000280	2.1000273	2.1000275	2.1000281	2.1000281	2.1000275	2.1000283	2.1000278	2.1000273	2.100027804

Hasil Pengujian Ukuran Populasi

Interval	Fitness Percobaan ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
10	2.1000164	2.1000204	2.1000157	2.1000200	2.1000193	2.1000109	2.1000124	2.1000073	2.1000201	2.1000258	2.100016829
40	2.1000251	2.1000270	2.1000283	2.1000275	2.1000249	2.1000244	2.1000270	2.1000250	2.1000252	2.1000250	2.100025939
70	2.1000247	2.1000256	2.1000281	2.1000269	2.1000276	2.1000270	2.1000279	2.1000276	2.1000251	2.1000274	2.100026782
100	2.1000280	2.1000272	2.1000266	2.1000269	2.1000257	2.1000280	2.1000276	2.1000272	2.1000263	2.1000273	2.100027086
130	2.1000265	2.1000266	2.1000274	2.1000279	2.1000278	2.1000281	2.1000280	2.1000283	2.1000271	2.1000275	2.100027531

Interval	Fitness Percobaan ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
160	2.1000281	2.1000268	2.1000273	2.1000280	2.1000278	2.1000278	2.1000283	2.1000283	2.1000278	2.1000283	2.100027842
190	2.1000279	2.1000274	2.1000267	2.1000285	2.1000279	2.1000270	2.1000284	2.1000277	2.1000282	2.1000282	2.100027774
220	2.1000280	2.1000283	2.1000277	2.1000283	2.1000282	2.1000276	2.1000279	2.1000284	2.1000279	2.1000276	2.100028004
250	2.1000274	2.1000282	2.1000271	2.1000282	2.1000273	2.1000285	2.1000279	2.1000278	2.1000272	2.1000284	2.100027811

Hasil Pengujian Banyak Generasi

Generasi	Fitness Percobaan ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
100	2.1000283	2.1000268	2.1000285	2.1000282	2.1000283	2.1000273	2.1000282	2.1000284	2.1000279	2.1000272	2.100027900
200	2.1000279	2.1000270	2.1000282	2.1000271	2.1000282	2.1000278	2.1000285	2.1000281	2.1000284	2.1000282	2.100027939
300	2.1000276	2.1000279	2.1000285	2.1000282	2.1000285	2.1000277	2.1000272	2.1000282	2.1000284	2.1000283	2.100028049
400	2.1000281	2.1000283	2.1000279	2.1000282	2.1000283	2.1000280	2.1000280	2.1000280	2.1000282	2.1000277	2.100028080
500	2.1000279	2.1000282	2.1000281	2.1000279	2.1000284	2.1000280	2.1000284	2.1000284	2.1000284	2.1000280	2.100028166
600	2.1000285	2.1000278	2.1000282	2.1000285	2.1000277	2.1000285	2.1000279	2.1000284	2.1000280	2.1000283	2.100028175
700	2.1000281	2.1000282	2.1000284	2.1000282	2.1000285	2.1000285	2.1000282	2.1000281	2.1000285	2.1000272	2.100028180
800	2.1000280	2.1000280	2.1000280	2.1000280	2.1000282	2.1000281	2.1000285	2.1000284	2.1000282	2.1000284	2.100028185
900	2.1000285	2.1000285	2.1000278	2.1000284	2.1000281	2.1000277	2.1000282	2.1000282	2.1000283	2.1000282	2.100028174
1000	2.1000283	2.1000283	2.1000283	2.1000281	2.1000285	2.1000284	2.1000281	2.1000281	2.1000280	2.1000278	2.100028189

Hasil Pengujian Kombinasi Nilai Cr dan Mr

Cr Mr	Fitness Percobaan ke-										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
0 1	2.1000272	2.1000278	2.1000255	2.1000243	2.1000258	2.1000267	2.1000264	2.1000263	2.1000268	2.1000249	2.100026152
0.1 0.9	2.1000285	2.1000276	2.1000270	2.1000266	2.1000283	2.1000266	2.1000253	2.1000273	2.1000276	2.1000262	2.100027098
0.2 0.8	2.1000273	2.1000283	2.1000283	2.1000280	2.1000284	2.1000276	2.1000267	2.1000275	2.1000285	2.1000282	2.100027881
0.3 0.7	2.1000285	2.1000284	2.1000284	2.1000274	2.1000283	2.1000285	2.1000278	2.1000281	2.1000283	2.1000276	2.100028126
0.4 0.6	2.1000283	2.1000285	2.1000274	2.1000285	2.1000284	2.1000278	2.1000285	2.1000283	2.1000280	2.1000285	2.100028207
0.5 0.5	2.1000282	2.1000285	2.1000284	2.1000281	2.1000276	2.1000279	2.1000282	2.1000283	2.1000284	2.1000285	2.100028193
0.6 0.4	2.1000282	2.1000283	2.1000281	2.1000274	2.1000285	2.1000284	2.1000285	2.1000284	2.1000284	2.1000285	2.100028260
0.7 0.3	2.1000285	2.1000282	2.1000279	2.1000285	2.1000282	2.1000276	2.1000278	2.1000282	2.1000279	2.1000282	2.100028106
0.8 0.2	2.1000267	2.1000275	2.1000281	2.1000275	2.1000281	2.1000271	2.1000282	2.1000283	2.1000277	2.1000282	2.100027744
0.9 0.1	2.1000279	2.1000285	2.1000275	2.1000281	2.1000279	2.1000284	2.1000277	2.1000271	2.1000278	2.1000273	2.100027824
1 0	2.1000280	2.1000276	2.1000274	2.1000278	2.1000279	2.1000278	2.1000271	2.1000273	2.1000271	2.1000277	2.100027591

