

OPTIMASI KOMPOSISI PAKAN KUDA DEWASA MENGUNAKAN ALGORITME GENETIKA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Rheza Raditya Andrianto

NIM: 135150201111168



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

PENGESAHAN

OPTIMASI KOMPOSISI PAKAN KUDA DEWASA MENGGUNAKAN ALGORITME
GENETIKA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rheza Raditya Andrianto
NIM : 135150201111168

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Januari 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Lailil Muflikah, S.Kom, M.Sc
NIP: 19741113 200501 2 001

Bayu Rahayudi, S.T, M.T
NIP: 19740712 200604 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

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

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, 15 Januari 2018



Rheza Raditya Andrianto

NIM: 135150201111168



KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa, atas berkat, rahmat, cinta kasih dan penyertaan-Nya penulis dapat menyelesaikan penelitian dan penyusunan naskah skripsi yang berjudul “Optimasi Komposisi Pakan Kuda Dewasa Menggunakan Algoritme Genetika”. Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Informatika/Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya Malang. Penulis menyadari bahwa dalam pembuatan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Lailil Muflikhah, S.Kom, M.Sc, dan Bayu Rahayudi S.T, M.T, selaku dosen Pembimbing I dan Pembimbing II yang telah banyak memberikan ilmu dan saran yang bermanfaat selama penyusunan skripsi ini.
2. Wayan Firdaus Mahmudy, S.Si, M.Kom, selaku Dekan, Heru Nurwasito, Ir., M.Kom, selaku Wakil Dekan I, Marji, Drs., M.T, selaku Wakil Dekan II, Edy Santoso, S.Si, M.Kom, selaku Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Tri Astoto Kurniawan, S.T, M.T, Ph.D, selaku Ketua Jurusan Teknik Informatika, Agus Wahyu Widodo, S.T, M.Cs, selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Dr. Wahyu Santoso, M. Sc., Ir. Bambang Edy Santoso, M.P., selaku dokter hewan dan ahli peternakan yang bersedia menjadi pakar untuk berkonsultasi tentang penyusunan komposisi pakan kuda dewasa.
5. Kedua orang tua penulis yang senantiasa mendukung dan memberi doa demi terselesaikannya skripsi ini.
6. Teman-teman seperjuangan Jumerlyanti, Yobel, Meirieska dan teman-teman informatika angkatan 2013 yang selalu memberi dukungan dan doa demi terselesaikannya skripsi ini, serta semua pihak yang tidak dapat penulis cantumkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Semoga segala bantuan yang telah diberikan kepada penulis selama penulis menyusun skripsi ini mendapat balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini tidak sempurna dan tidak luput dari kesalahan, sehingga penulis menerima apabila terdapat kritik dan saran. Semoga skripsi ini dapat bermanfaat.

Malang, 15 Januari 2018

Penulis
rhezara.rr@gmail.com

ABSTRAK

Rheza Raditya Andrianto. 2018. Optimasi Pakan Kuda Dewasa Menggunakan Algoritme Genetika. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Lailil Muflikhah, S.Kom, M.Sc dan Bayu Rahayudi S.T, M.T.

Indonesia merupakan negara yang memiliki keberagaman populasi ternak diantaranya sapi, kambing, domba, dan kuda. Seiring dengan perkembangan zaman yang semakin modern membuat populasi kuda sendiri mengalami penurunan karena pudarnya peranan kuda dalam berbagai hal serta pemberian pakan yang tidak sesuai dengan standar nutrisi. Pada penelitian ini mengimplementasikan algoritme genetika untuk mendapatkan komposisi pakan dengan biaya yang terjangkau namun standar nutrisi yang diperlukan tetap terpenuhi sehingga mampu menjaga kesehatan dan kestabilan kuda guna kelancaran berkembang biak. Representasi yang digunakan pada penelitian ini adalah *real code* dimana setiap kromosom menginisialisasi bahan pakan yang digunakan. Metode *crossover* yang digunakan yaitu *extended intermediate*. Metode mutasi yang digunakan yaitu *reciprocal exchange mutation*, dan metode seleksi menggunakan metode *elitism selection*. Berdasarkan hasil pengujian yang telah dilakukan, didapatkan parameter optimal yaitu pada populasi 70, generasi 250 serta kombinasi *crossover rate* dan *mutation rate* sebesar 0,5 dan 0,5 dengan *fitness* sebesar 0,18887407184772886. Hasil akhir yang didapatkan berupa komposisi pakan dengan harga yang minimal berdasarkan kebutuhan nutrisi kuda dewasa.

Kata kunci: Algoritme genetika, kuda, komposisi pakan

ABSTRACT

Rheza Raditya Andrianto. 2018. *Optimization of Adult Horse Feed Using Genetic Algorithm. Faculty of Computer Science, Brawijaya University, Malang. Lecturer: Lailil Muflikhah, S.Kom, M.Sc dan Bayu Rahayudi S.T, M.T.*

Indonesia is a country that has a diversity of livestock population such as cows, goats, sheep, and horses. Over time followed by the modern era, the horse population decreased because of the role of horses in various ways and non-fulfillment of nutrition by the horses feed. In this study tries to implement genetic algorithm to get the composition of feed at a minimal cost but the required nutritional standards remain met so as to maintain the health and stability of the horse breed. The representation used in this study is a real code in which each chromosome initializes the feed ingredients used. The crossover method used is the extended intermediate. The mutation method used is reciprocal exchange mutation, and selection method used is elitism selection. Based on the results of this study, obtained the optimal parameters at 70 population, 250 generation and combination of crossover rate and mutation rate as 0.5 and 0.5 with the highest fitness 0.18887407184772886. The result obtained in the form of feed composition with a minimum cost based on the nutritional needs of adult horses.

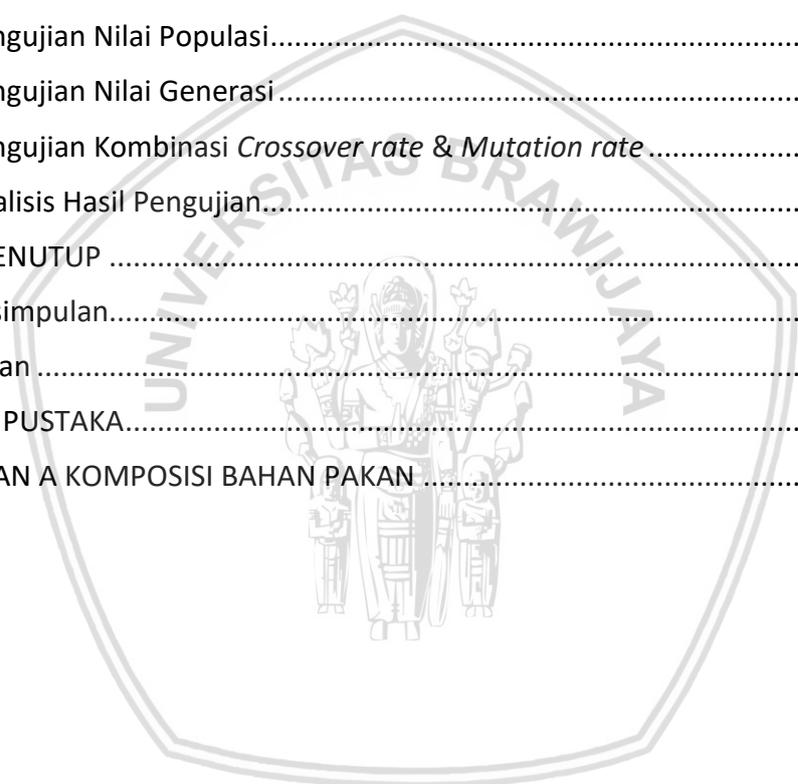
Keywords: Genetic algorithm, horse, feed composition

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR <i>SOURCE CODE</i>	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 Kuda	8
2.3 Pakan Ternak	8
2.3.1 Serat.....	9
2.3.2 Konsentrat	9
2.4 Nutrisi Pakan Kuda.....	10
2.4.1 Energi Metabolis.....	11
2.4.2 Protein	12
2.4.3 Mineral	12
2.4.4 Vitamin	12
2.4.5 Air	12
2.5 Algoritme Genetika.....	13
2.5.1 Representasi Kromosom	13

2.5.2 Inisialisasi Populasi	13
2.5.3 Crossover	14
2.5.4 Mutasi.....	14
2.5.5 Evaluasi.....	14
2.5.6 Seleksi	15
BAB 3 METODOLOGI	16
3.1 Studi Literatur	16
3.2 Pengumpulan Data	17
3.3 Analisa Kebutuhan	17
3.4 Perancangan Sistem.....	17
3.5 Implementasi Sistem	17
3.6 Pengujian dan Analisis	17
3.7 Kesimpulan dan Saran	18
BAB 4 PERANCANGAN.....	19
4.1 Formulasi Permasalahan.....	19
4.2 Siklus Penyelesaian Masalah Menggunakan Algoritme Genetika.....	20
4.2.1 Representasi Kromosom dan Perhitungan <i>Fitness</i>	21
4.2.2 Inisialisasi Populasi Awal	27
4.2.3 Reproduksi.....	28
4.2.4 Evaluasi.....	31
4.2.5 Seleksi	33
4.3 Perancangan Antarmuka	34
4.3.1 Rancangan Tampilan Halaman <i>Input</i> Data Kuda dan Bahan Pakan.....	34
4.3.2 Rancangan Tampilan Halaman Algoritme Genetika	35
4.4 Perancangan Pengujian Sistem.....	35
4.4.1 Pengujian Nilai Populasi	36
4.4.2 Pengujian Nilai Generasi	36
4.4.3 Pengujian Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	37
4.4.4 Pengujian Validasi dengan Parameter Terbaik	37
BAB 5 IMPLEMENTASI	39
5.1 Implementasi Program	39
5.1.1 Implementasi Inisialisasi Populasi Awal	39

5.1.2 Implementasi Proses <i>Crossover</i>	39
5.1.3 Implementasi Proses Mutasi	40
5.1.4 Implementasi Perhitungan Nilai <i>Fitness</i>	41
5.1.5 Implementasi Proses Seleksi	43
5.2 Implementasi Antarmuka	44
5.2.1 Implementasi Halaman <i>Input</i> Data Kuda dan Bahan Pakan	44
5.2.2 Implementasi Halaman Algoritme Genetika	45
BAB 6 PENGUJIAN	46
6.1 Sistematika Pengujian.....	46
6.2 Pengujian Nilai Populasi.....	46
6.3 Pengujian Nilai Generasi	47
6.4 Pengujian Kombinasi <i>Crossover rate</i> & <i>Mutation rate</i>	49
6.5 Analisis Hasil Pengujian.....	50
BAB 7 PENUTUP	53
7.1 Kesimpulan.....	53
7.2 Saran	53
DAFTAR PUSTAKA.....	55
LAMPIRAN A KOMPOSISI BAHAN PAKAN	57



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	6
Tabel 2.2 Kebutuhan Nutrisi Pakan Kuda.....	10
Tabel 2.3 Contoh Representasi kromosom.....	13
Tabel 2.5 Contoh <i>Reciprocal Exchange Mutation</i>	14
Tabel 4.1 Contoh Representasi Kromosom	21
Tabel 4.2 Perhitungan Bobot Bahan Pakan P1	22
Tabel 4.3 Rincian Data Bahan Pakan.....	22
Tabel 4.4 Kandungan Nutrisi Bahan Pakan	22
Tabel 4.5 Kebutuhan dan Ketersediaan Nutrisi	25
Tabel 4.6 Total Harga Individu P1	26
Tabel 4.7 Nilai <i>fitness</i> P1	27
Tabel 4.8 Pembangkitan Populasi Awal	27
Tabel 4.9 <i>Parent Crossover</i> Terpilih	30
Tabel 4.10 Membangkitkan Nilai α	30
Tabel 4.11 Hasil <i>Offspring Crossover</i>	30
Tabel 4.12 <i>Parent</i> Mutasi Terpilih.....	31
Tabel 4.13 Hasil <i>Offspring</i> Mutasi	31
Tabel 4.14 Hasil Evaluasi	32
Tabel 4.15 Hasil Seleksi	34
Tabel 4.16 Rancangan Pengujian Populasi.....	36
Tabel 4.17 Rancangan Pengujian Generasi	36
Tabel 4. 18 Rancangan Pengujian Kombinasi <i>crossover rate</i> dan <i>mutation rate</i> .	37
Tabel 6.1 Pengujian Nilai Populasi	46
Tabel 6.2 Pengujian Nilai Generasi	48
Tabel 6.3 Pengujian Kombinasi <i>Crossover rate</i> & <i>Mutation rate</i>	49
Tabel 6.4 Pengujian Komposisi 1 dengan Parameter Terbaik	51
Tabel 6.5 Pengujian Komposisi 2 dengan Parameter Terbaik	51
Tabel 6.6 Pengujian Komposisi 3 dengan Parameter Terbaik	52
Tabel 6.7 Pengujian Komposisi 3 dengan Parameter Terbaik	52

DAFTAR GAMBAR

Gambar 2.1 Contoh Jenis Rumpit	9
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	16
Gambar 4.1 <i>Flowchart</i> Algoritme Genetika	20
Gambar 4.3 <i>Flowchart</i> Pembangkitan Populasi Awal	28
Gambar 4.4 <i>Flowchart</i> Proses <i>Crossover</i>	29
Gambar 4.5 <i>Flowchart</i> Proses Mutasi	31
Gambar 4.2 <i>Flowchart</i> Proses Hitung <i>Fitness</i>	32
Gambar 4.6 <i>Flowchart</i> Proses Seleksi	33
Gambar 4.7 Rancangan Tampilan Halaman <i>Input</i> Data Kuda dan Bahan Pakan..	34
Gambar 4.8 Rancangan Tampilan Halaman Algoritme Genetika	35
Gambar 5.1 Halaman <i>Input</i> Data Kuda dan Bahan Pakan	45
Gambar 5.2 Halaman Algoritme Genetika	45
Gambar 6.1 Pengujian Nilai Populasi	47
Gambar 6.2 Pengujian Nilai Generasi	48
Gambar 6.3 Pengujian Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i>	50

DAFTAR SOURCE CODE

<i>Source Code</i> 5.1 Inialisasi Populasi Awal	39
<i>Source Code</i> 5.2 Proses <i>Crossover</i>	40
<i>Source Code</i> 5.3 Proses Mutasi	41
<i>Source Code</i> 5.4 Perhitungan Nilai <i>Fitness</i>	42
<i>Source Code</i> 5.5 Proses Seleksi	44



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia memiliki keragaman ternak dengan tingkat populasi yang mengalami peningkatan, namun hal berbeda terjadi pada populasi kuda (Mansyur, 2006). Turunnya populasi ini disebabkan oleh pudarnya peranan kuda dalam sektor pertanian baik sebagai penghasil daging maupun tenaga kerja. Selain itu, peranannya sebagai alat transportasi dan militer juga kian berkurang. Oleh karena itu, pengembangan peternakan kuda saat ini cenderung lebih ke arah olahraga dan rekreasi sehingga pemeliharannya pun harus lebih intensif (Maswarni & Rachman, 2014). Kuda pada dasarnya adalah hewan yang hidup berpindah tempat dan memiliki semangat tinggi. Di alam liar tingkat efisiensi reproduksi pada kuda mampu melebihi 90% namun tingkat efisiensi reproduksi tersebut mengalami penurunan apabila kuda dijadikan hewan ternak akibat kurangnya latihan fisik, gangguan kesehatan, serta kebutuhan pakan yang kurang tercukupi menyebabkan rendahnya tingkat konsepsi/kebuntingan serta rendahnya kelahiran (Rahmawati, 2011).

Pakan merupakan salah satu faktor penting dalam dunia ternak. Setiap hewan ternak akan membutuhkan nutrisi pakan yang sesuai dengan standar kebutuhan nutrisi dari ternak tersebut. Kekurangan nutrisi pakan yang terjadi dalam waktu lama akan menyebabkan gangguan reproduksi yang parah disertai dengan pertumbuhan yang lambat pada ternak. Pemberian pakan ternak tanpa memperhatikan kualitas dan kuantitas akan gangguan kesehatan sehingga menurunkan tingkat efisiensi reproduksi ternak (Marginingtyas, 2015).

Menurut (Maswarni & Rachman, 2014) pemberian pakan pada tingkatan yang tepat dapat menghindari kelebihan atau kekurangan nutrisi. Tingkatan pemberian pakan pada kuda dewasa diklasifikasikan menjadi pakan untuk induk kuda, pakan kuda bunting, pakan kuda menyusui fase awal, pakan kuda menyusui fase kedua, pakan kuda pacu, dan pakan kuda pejantan. Setiap tingkatan memiliki kebutuhan nutrisi yang bervariasi. Sehingga pemberian pakan terhadap kuda dewasa harus disesuaikan terhadap tingkatannya.

Kebutuhan zat nutrisi utama yang dibutuhkan dalam pakan ternak kuda dewasa berupa ME (Energi Metabolis), protein, mineral, vitamin, dan air (Maswarni & Rachman, 2014). Pada umumnya pakan kuda dibagi menjadi 2 kategori yaitu serat dan konsentrat. Serat didapatkan dari rumput-rumputan, sedangkan konsentrat adalah pakan yang mengandung unsur protein, karbohidrat, lemak dan mineral. Pemberian kedua pakan ini haruslah seimbang. Dalam proses pemberian pakan, kurangnya informasi mengenai kebutuhan nutrisi kuda yang dimiliki oleh para peternak menyebabkan hilangnya keseimbangan nutrisi yang harusnya terpenuhi. Untuk itu mengoptimalkan pemberian pakan kuda dewasa merupakan cara untuk membantu peternak dalam memenuhi kecukupan nutrisi yang dibutuhkan serta menekan biaya pemenuhan kebutuhan

pakan, dimana diketahui bahwa biaya pembelian pakan ternak merupakan biaya tertinggi dalam menjalankan usaha peternakan (Nugraha, 2011).

Untuk pengoptimalan pemberian pakan pada kuda dewasa dapat dilakukan dengan cara mengkombinasikan jenis-jenis pakan yang saat dicampur akan memenuhi kebutuhan nutrisi kuda dewasa namun tetap memiliki harga yang rendah. Algoritme genetika merupakan salah satu algoritme evolusi yang paling populer karena terbukti mampu digunakan untuk memberikan solusi terhadap berbagai masalah yang kompleks dalam berbagai (Mahmudy, 2013).

Menyelesaikan masalah optimasi komposisi bahan pakan sudah pernah dilakukan sebelumnya. Fakhroh (2015) mengimplementasikan algoritme genetika dalam mengoptimasikan komposisi bahan pakan pada sapi perah untuk memenuhi kebutuhan nutrisi dengan harga kecil. Dari hasil penelitian didapatkan hasil optimal yang dicapai dari ukuran populasi berada di ukuran populasi ke 100, jumlah generasi 200 serta kombinasi *crossover rate* dan *mutation rate* 0,3 dan 0,3. Kromosom terbaik yang diperoleh sudah bisa menekan biaya dan memaksimalkan kebutuhan nutrisi. Penelitian lainnya oleh Saraswati (2016) mengenai optimasi komposisi pakan ikan patin menggunakan algoritme genetika. Didapatkan solusi yang paling optimal dari ukuran populasi 90, jumlah generasi 300 serta kombinasi *crossover rate* dan *mutation rate* 0,5 dan 0,5. Kromosom terbaik yang diperoleh memberikan solusi komposisi pakan terbaik untuk ikan patin dengan harga yang minimum.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah di paparkan, maka dibentuk rumusan masalah yang dibahas pada penelitian skripsi ini adalah:

1. Bagaimana implementasi algoritme genetika dalam optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika?
2. Bagaimana menentukan kombinasi parameter algoritme genetika yang tepat dalam menyusun komposisi jumlah pakan kuda dewasa?
3. Bagaimana mengukur tingkat kualitas dari solusi yang diberikan algoritma genetika terhadap optimasi komposisi pakan kuda dewasa?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dari penelitian skripsi ini adalah sebagai berikut:

1. Mengimplementasikan algoritme genetika dalam optimasi komposisi pakan kuda dewasa.
2. Mengetahui kombinasi parameter algoritme genetika yang tepat untuk optimasi komposisi pakan kuda dewasa.
3. Mengetahui tingkat kualitas solusi yang diberikan algoritme genetika terhadap optimasi komposisi pakan kuda dewasa.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian skripsi ini adalah untuk mendapatkan solusi pemberian pakan bagi kuda dewasa secara optimal sesuai standar kebutuhan nutrisi dan menekan biaya yang diperlukan.

1.5 Batasan masalah

Batasan masalah dalam penelitian ini ditunjukkan untuk menghindari pelebaran masalah yang akan dibahas, berikut adalah batasan-batasannya:

1. Acuan nutrisi yang digunakan berupa ME (Energi Metabolis), protein, mineral kalsium, dan fosfor.
2. Terdapat 8 jenis berat badan dan 4 fase kuda dewasa yang akan digunakan yaitu pemeliharaan, 90 hari sebelum melahirkan, menyusui 3 bulan awal, menyusui 3 bulan akhir.
3. Terdapat 6 kombinasi jenis pakan yang digunakan terdiri dari 2 pakan serat berbeda dan 4 pakan konsentrat berbeda berdasarkan stok pakan yang tersedia.
4. Harga pakan yang digunakan mengacu pada informasi yang didapatkan melalui observasi pada beberapa tempat penjualan pakan ternak di kota Batu pada bulan Februari 2017.
5. Hasil rekomendasi pakan yang diberikan sistem hanya untuk tiap satu kuda dengan pemberian sekali pakan dan fase serta bobot yang dipilih.
6. Tidak dilakukan perbandingan terhadap metode lain.

1.6 Sistematika pembahasan

Memberi uraian tentang latar belakang, rumusan masalah, tujuan dan manfaat yang ingin dicapai, batasan masalah, serta sistematika pembahasan penulisan laporan penelitian optimasi komposisi pakan kuda dewasa.

BAB 2 Landasan Kepustakaan

Memberi uraian tentang kajian pustaka atau berbagai referensi pendukung yang berkaitan dengan kuda, dan algoritme genetika serta teori untuk menyelesaikan permasalahan.

BAB 3 Metodologi

Memberi uraian tentang metode yang digunakan untuk penelitian yang terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan pengambilan kesimpulan.

BAB 4 Perancangan

Memberi uraian tentang perancangan sistem optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika yang akan dibangun meliputi formulasi permasalahan, siklus penyelesaian masalah, perancangan antarmuka, serta perancangan pengujian sistem.

BAB 5 Implementasi

Memberi uraian tentang Implementasi program optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika dan antarmuka berdasarkan perancangan yang telah dibuat.

BAB 6 Pengujian

Berisi uraian tentang pengujian dan analisis hasil pengujian dari sistem optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika yang telah di implementasikan .

BAB 7 Penutup

Bab ini memuat kesimpulan yang dibuat berdasarkan perancangan hingga pengujian sistem yang dirancang serta saran-saran untuk pengembangan sistem yang lebih baik kedepannya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini mengacu pada penelitian sebelumnya yang menerapkan metode algoritme genetika untuk menyelesaikan permasalahan berbagai objek yang berbeda.

Penelitian oleh Fakhroh, et. al pada tahun 2014 yang menerapkan algoritme genetika untuk optimasi komposisi pakan sapi perah dengan tujuan membantu para peternak untuk mendapatkan komposisi bahan pakan yang sesuai dengan standar kebutuhan nutrisi sapi perah dengan biaya minimal agar memberikan kualitas susu hasil perah yang baik. Data pakan sapi yang dioptimasi terdapat 30 jenis dengan parameter nutrisi berupa Prdd (Protein dapat dicerna) dan MP (Martabat Pati). Hasil keluaran yang akan diperoleh adalah hasil rekomendasi komposisi pakan sapi perah dengan biaya yang minimal. Solusi yang optimal didapatkan pada populasi sebanyak 100 dan banyaknya generasi 200 serta kombinasi *crossover rate* dan *mutation rate* yaitu 0,7 dan 0,3 (Fakhroh, et. al., 2014).

Penelitian oleh Pramesti, et. al pada tahun 2014 yang menerapkan algoritme genetika untuk optimasi komposisi pakan kambing potong yaitu bertujuan untuk mengoptimalkan komposisi ransum pakan kambing potong untuk membantu peternak kambing potong menentukan komposisi pakan kambing yang sesuai kebutuhan nutrisi dengan biaya minimum. Hasil keluaran yang akan diperoleh adalah hasil rekomendasi pakan kambing potong dengan biaya minimal serta nutrisi yang didapatkan. Solusi yang optimal didapatkan pada populasi sebanyak 200 dan banyaknya generasi 200 dengan nilai kombinasi *crossover rate* dan *mutation rate* yaitu 0,1 dan 0,5 (Pramesti, et. al., 2014).

Penelitian oleh Wahid, et. al pada tahun 2015 yang menerapkan algoritme genetika untuk optimasi komposisi makanan bagi penderita kolesterol yaitu bertujuan untuk pengendalian kadar lemak bagi penderita kolesterol dari makanan dengan biaya minimal. Hasil keluaran yang akan diperoleh adalah bahan makanan bagi penderita kolesterol dengan kadar lemak yang rendah serta biaya yang minimum. Solusi yang optimal didapatkan pada populasi sebanyak 100 dan banyaknya generasi 90 dengan nilai kombinasi *crossover rate* dan *mutation rate* yaitu 0,7 dan 0,3 (Wahid, et. al., 2015).

Penelitian oleh Saraswati, et. al pada tahun 2016 yang menerapkan algoritme genetika untuk optimasi pakan ikan patin yaitu bertujuan untuk mengoptimalkan komposisi pakan ikan patin untuk menekan biaya produksi dalam hal penggunaan pakan namun kebutuhan nutrisi tetap terpenuhi. Hasil keluaran yang akan diperoleh setelah mendapatkan kromosom terbaik adalah hasil rekomendasi pakan ikan patin dengan biaya minimal serta nutrisi yang didapatkan. Solusi yang optimal didapatkan pada populasi sebanyak 90 dan banyaknya generasi 300 dengan nilai kombinasi *cr* dan *mr* yaitu 0,5 dan 0,5 (Saraswati, et. al., 2016).

Tabel 2.1 Kajian Pustaka

No.	Peneliti	Objek (Input)	Metode	Hasil (Output)
1.	Fakhroh, et. al., 2014	Objek: Pakan Sapi Perah Parameter Input: <ul style="list-style-type: none"> • Populasi • Generasi • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 	Metode : Algoritme Genetika Langkah-langkah: <ul style="list-style-type: none"> • Pembangkitan Populasi Awal • <i>Crossover</i> • Mutasi • Menghitung Nilai Fitness • Seleksi • Populasi Baru • Kromosom terbaik 	Rekomendasi komposisi pakan sapi perah dengan biaya yang minimal. Parameter terbaik <ul style="list-style-type: none"> • Populasi : 100 • Generasi : 200 • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 0,7 & 0,3
2.	Pramesti, et. al., 2014	Objek: Pakan Kambing Potong Parameter Input: <ul style="list-style-type: none"> • Jenis bahan pakan yang tersedia • Harga bahan pakan • Populasi • Generasi • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 	Metode : Algoritme Genetika Langkah-langkah: <ul style="list-style-type: none"> • Generasi Populasi Awal • <i>Crossover</i> • Mutasi • Fitness • Seleksi • Populasi baru • Kromosom terbaik 	Rekomendasi pakan kambing potong dengan biaya minimal serta nutrisi yang didapatkan. Parameter terbaik <ul style="list-style-type: none"> • Populasi : 200 • Generasi : 200 • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 0,1 & 0,5

3.	Wahid, 2015	<p>Objek: Makanan untuk penderita kolesterol</p> <p>Parameter Input:</p> <ul style="list-style-type: none"> • Jenis bahan makanan yang tersedia • Harga bahan makanan • Populasi • Generasi • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 	<p>Metode : Algoritme Genetika</p> <p>Langkah-langkah:</p> <ul style="list-style-type: none"> • Generasi Populasi Awal • <i>Crossover</i> • Mutasi • Fitness • Seleksi • Populasi baru • Kromosom terbaik 	<p>Rekomendasi makanan untuk penderita kolesterol dengan biaya yang minimum.</p> <p>Parameter terbaik</p> <ul style="list-style-type: none"> • Populasi : 100 • Generasi : 90 • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 0,7 & 0,3
4.	Saraswati, et. al., 2016	<p>Objek: Pakan Ikan Patin</p> <p>Parameter Input:</p> <ul style="list-style-type: none"> • Jenis bahan pakan yang tersedia • Harga bahan pakan • Populasi • Generasi • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 	<p>Metode : Algoritme Genetika</p> <p>Langkah-langkah:</p> <ul style="list-style-type: none"> • Generasi Populasi Awal • <i>Crossover</i> • Mutasi • Fitness • Seleksi • Populasi baru • Kromosom terbaik 	<p>Rekomendasi pakan ikan patin dengan biaya minimal serta nutrisi yang didapatkan.</p> <p>Parameter terbaik</p> <ul style="list-style-type: none"> • Populasi : 90 • Generasi : 300 • Kombinasi <i>Crossover rate</i> dan <i>Mutation rate</i> 0,5 & 0,5

2.2 Kuda

Kuda (*Equus caballus*) telah dikenal banyak orang sebagai hewan yang memiliki banyak fungsi seperti sebagai hewan peliharaan, olahraga ataupun sebagai sarana transportasi.

Menurut Maswarni & Rachman (2014), jenis kuda diklasifikasikan berdasarkan ukuran tinggi, berat, dan kegunaannya, yaitu:

1. Kuda Ringan (*Light Horses*)

Kuda ringan memiliki tulang-belulang yang kecil, kakinya tipis, dan memiliki berat sekitar 450-600 kg saat dewasa dan tinggi 146-173 cm. Kuda *Light Horses* umumnya lebih lincah sehingga banyak digunakan sebagai kuda pacu, kuda tunggang, dan membantu dalam peternakan.

2. Kuda Berat (*Heavy Horses*)

Kuda berat memiliki tulang-belulang yang besar, kakinya tebal, kuat dan mampu menjangkau berat lebih dari 700 kg saat dewasa dengan tinggi 147-157 cm. Dengan struktur tubuh yang kuat, kuda ini dimanfaatkan untuk menarik beban, kuda tunggang, dan untuk pekerjaan berat lainnya.

3. Kuda Poni dan Keledai

Kuda poni dan Keledai memiliki berat sekitar 250 kg hingga 450 kg saat dewasa dan tinggi di bawah 147 cm dengan punggung yang relatif pendek.

2.3 Pakan Ternak

Berdasarkan spesiesnya, kuda merupakan hewan mamalia namun kebutuhan gizi yang diperlukan kuda tidaklah sama dengan kebutuhan hewan ternak mamalia lainnya seperti sapi atau kambing. Meskipun tergolong hewan herbivora yang memakan rumput dan jerami namun kuda tidak termasuk ruminansia (memiliki 4 lambung) dan tidak dapat memamah makanan yang telah dicerna atau regurgitasi dan menyebabkan kuda kurang efisien dalam memanfaatkan pakan berserat (Rahmawati, 2011). Sehingga peternak perlu menyediakan pakan berupa konsentrat. Untuk menentukan jumlah komposisi pakan yang seimbang antara serat dan konsentrat, peneliti menggunakan persamaan jumlah ideal pakan kuda perhari berdasarkan persentase bobot kuda (Hammer, 1993). Persamaan jumlah ideal pakan kuda ditunjukkan pada Persamaan 2.1.

$$P = 2\% \times B(kg) \quad (2.1)$$

Jumlah pakan ideal tersebut harus disesuaikan dengan kebutuhan ideal serat dan konsentrat dari kuda yang ditunjukkan pada Persamaan 2.2.

$$\begin{aligned} S_{Min} &= 1\% \times B(kg) \\ K_{Min} &= 0.5\% \times B(kg) \end{aligned} \quad (2.2)$$

Keterangan:

P = Jumlah pakan maksimum kuda perhari

$B(kg)$ = Bobot kuda dalam satuan kilogram

$SMin$ = Jumlah serat minimum yang diperlukan dalam pakan

$KMin$ = Jumlah konsentrat minimum yang diperlukan dalam pakan

2.3.1 Serat

Serat merupakan bagian penting dalam pakan kuda karena mempengaruhi kesehatan pencernaan pada kuda. Bahan kasar yang dimiliki oleh serat membantu proses transportasi dan pemecahan bahan konsentrat dalam proses pencernaan. Di Indonesia sendiri terdapat beberapa jenis sumber serat yang digunakan sebagai pakan kuda, antara lain rumput *panicum muticum* dan *braccaria mutica* (Soeharjono, 1990). Kuda yang tidak menerima asupan serat yang memadai tidak hanya meningkatkan risiko gangguan pencernaan, tetapi juga akan mengakibatkan masalah perilaku. Pada umumnya kebutuhan serat pada kuda dapat diperoleh dari hijauan yaitu rumput. Di Indonesia sendiri jenis rumput yang biasa dijadikan bahan pakan pada kuda yaitu rumput kolonjono (*Panicum muticum*) dan rumput malela (*Brachiaria mutica*).



Gambar 2.1 Contoh Jenis Rumput

2.3.2 Konsentrat

Konsentrat merupakan pakan yang dipadukan bersama bahan pakan lain untuk mendapatkan keserasian nutrisi dari pakan. Peranan konsentrat adalah untuk meningkatkan nilai nutrisi yang rendah agar memenuhi kebutuhan normal hewan untuk tumbuh dan berkembang secara sehat (Akoso, 1996). Konsentrat bagi pakan ternak dapat diperoleh dari biji-bijian, jagung, tepung kedelai, menir, dedak, bekatul, bungkil kelapa, dan umbi. Penambahan konsentrat pada pakan sebagai sumber protein, karbohidrat, lemak, dan mineral dapat meningkatkan daya cerna, pertambahan bobot badan serta efisien dalam penggunaan ransum (Holcomb et. al., 1984).

Berdasarkan kandungan gizinya, konsentrat dibagi dua golongan, yaitu (Wahiduddin, 2008):

1. Proteinaseous

Proteinaseous yaitu konsentrat sebagai sumber protein apabila kandungan protein lebih dari 18%, *Total Digestible Nutrision* (TDN) 60%. Jenis konsentrat ini dapat diperoleh dari hewan dan tumbuhan. Berasal dari hewan

mengandung protein lebih dari 47%. Mineral kalsium (Ca) lebih dari 1% dan fosfor (P) lebih dari 1,5% serta kandungan serat kasar di bawah 2,5%. Contohnya tepung ikan, tepung susu, tepung daging, tepung darah, tepung bulu dan tepung cacing. Berasal dari tumbuhan, kandungan proteinnya di bawah 47%, mineral kalsium (Ca) di bawah 1% dan fosfor (P) di bawah 1,5% serat kasar lebih dari 2,5%. Contohnya tepung kedelai, tepung biji kapuk, tepung bunga matahari, bungkil wijen, bungkil kedelai, bungkil kelapa, dan bungkil kelapa sawit.

2. *Carbonaceous*

Carbonaceous yaitu konsentrat sebagai sumber energi apabila kandungan protein di bawah 18%, *Total Digestible Nutrition* (TDN) 60% dan serat kasarnya lebih dari 10%. Contohnya dedak, jagung, empok, dan polar.

2.4 Nutrisi Pakan Kuda

Kuda membutuhkan sejumlah nutrisi yang lengkap untuk menunjang kebutuhan pokok yaitu bernapas, bergerak, dan aktifitas lain berdasarkan peranan kuda serta memberikan efisiensi terhadap pertumbuhan dan produktifitas dari kuda. Nutrisi yang dimaksud berupa energi metabolis, protein, mineral, vitamin, dan air. Pemberian asupan nutrisi terhadap kuda dipengaruhi oleh tingkatan kuda, sehingga dibutuhkan keseimbangan susunan antara serat dan konsentrat yang menjadi pakan kuda untuk memperoleh nutrisi yang tepat dan seimbang. Dalam penelitian ini digunakan kuda dewasa yang memasuki tahap usia reproduksi dan kuda yang sedang bunting, serta saat fase menyusui. Detail nutrisi yang dibutuhkan akan ditunjukkan pada Tabel 2.2

Tabel 2.2 Kebutuhan Nutrisi Pakan Kuda

Berat Badan (kg)	Digestible Energy (Mcal)	Protein (gr)	Calsium (gr)	Fosfor (gr)
Pemeliharaan				
200	6.7	252	8	5.6
300	10	378	12	8.4
400	13.3	504	16	11.2
500	16.7	630	20	14
600	20	756	24	16.8
700	23.4	883	28	19.6
800	26.7	1008	32	22.4
900	30.0	1134	36	25.2
90 Hari Sebelum Melahirkan				
200	8.2	338	14.4	10.5
300	12.2	506	21.6	15.7



Berat Badan (kg)	Digestible Energy (Mcal)	Protein (gr)	Calcium (gr)	Fosfor (gr)
400	16.2	675	28.8	21
500	20.3	844	36	26.3
600	24.3	1012	43.2	31.5
700	28.4	1180	50.4	36.8
800	32.4	1348	57.6	42
900	36.5	1520	64.8	47.3
Menyusui, 3 Bulan Awal				
200	12.5	604	23.2	15
300	18.8	906	34.8	22.5
400	25	1208	46.4	30
500	31.3	1511	58	37.5
600	37.6	1813	69.6	45
700	43.9	2115	81.2	52.5
800	50.2	2417	92.8	60
900	56.5	2719	104.3	67.5
Menyusui, 3 Bulan Akhir				
200	11.3	532	15.8	9.9
300	17	798	23.7	14.9
400	22.7	1064	31.6	19.7
500	28.3	1331	39.5	24.7
600	34	1597	47.4	29.6
700	39.7	1863	55.3	34.6
800	45.4	2129	63.2	39.6
900	51.1	2395	71.2	44.5

Sumber: *Nutrient Requirements of Horses* (2007)

2.4.1 Energi Metabolis

Energi metabolis pada kuda bersumber utama dari karbohidrat yang diperoleh dari pakan konsentrat. Energi yang dihasilkan akan digunakan untuk keseluruhan aktifitas di dalam tubuh, misalnya seperti proses pencernaan, pernapasan, reproduksi, dan lain-lain (Marginingtyas, 2015). Jika kuda kekurangan karbohidrat sebagai sumber energi utama, maka lemak akan berperan sebagai cadangan energi. Apabila lemak masih tidak mampu mencukupi kebutuhan energi, maka protein akan dirombak menjadi energi (Sudarmono, 2003).

Konsentrat dengan kandungan energi metabolis yang tinggi akan menyebabkan kuda kurang mengkonsumsi pakan sehingga kuda akan kekurangan zat gizi lain yang diperlukan. Sehingga pemberian pakan dengan kandungan energi metabolis harus disesuaikan dengan tingkatannya.

2.4.2 Protein

Protein merupakan komponen organik kompleks yang terdiri dari karbon, hidrogen, oksigen, sulfur, serta fosfor dan zat besi. Secara umum fungsi protein dalam tubuh adalah untuk pertumbuhan dan reproduksi, akan tetapi pada ternak kuda dengan tujuan pemeliharaan untuk dipacu maka kebutuhan protein untuk kuda pacu adalah untuk menjaga keseimbangan otot, kerangka dan sistem saraf serta untuk pembentukan kulit dan rambut (Tulung, 2012).

Kebutuhan untuk protein dapat terpenuhi dari bahan makanan yang mengandung protein nabati maupun hewani, yaitu (Marginingtyas, 2015):

1. Protein nabati : jagung, bungkil kacang kedelai, bungkil kelapa, bungkil kacang tanah, dll.
2. Protein hewani : tepung ikan, tepung kerang, tepung susu, dll.

2.4.3 Mineral

Mineral adalah bahan anorganik penting yang harus hadir dalam jumlah yang cukup untuk fungsi tubuh dengan benar. Mineral ini termasuk kalsium, fosfor, natrium, dan klorida, dan biasanya ditemukan pakan serat yang bersumber dari rumput. Kalsium dibutuhkan sebagai komponen esensial dalam pembentukan tulang, tulang rawan, serta pemeliharaan dan fungsi kerangka, saraf, dan otot. Kuda akan kehilangan mineral yang dikeluarkan oleh keringat sehingga kebutuhan mineral mengacu pada aktifitas kuda. Dan bagi kuda yang menjalani aktifitas berat sebaiknya diberikan tambahan asupan suplemen elektrolit sebanyak 1-4 sendok makan sehari ke dalam air minum serta tambahan yang berbentuk blok garam (Tulung, 2012).

2.4.4 Vitamin

Vitamin merupakan senyawa esensial yang dapat diperoleh dari luar tubuh. Vitamin pada kuda berfungsi untuk proses metabolisme zat-zat makanan serta memaksimalkan stamina dan fungsi otot pada kuda (Tulung, 2012). Sebagian besar kebutuhan vitamin kuda sudah terpenuhi melalui pakan hijauan, sehingga kualitas hijauan yang baik akan mencegah dibutuhkannya penambahan asupan vitamin sebagai suplemen pada kuda.

2.4.5 Air

Tubuh hewan terdiri dari 60-70% air. Pada umumnya kuda membutuhkan air minum sekitar 40-45 liter perhari. Beberapa elemen mineral akan hilang bersamaan dengan urine dan keringat. Sehingga penggantian air dan elektrolit merupakan hal penting untuk mencegah kelelahan, kerusakan otot, dan berkurangnya nafsu makan. Air bersih harus tersedia setiap saat agar kebutuhan

air terpenuhi dan kuda lebih rentan terhadap kondisi dehidrasi, *impactions usus*, dan bentuk lain dari kolik (Tulung, 2012).

2.5 Algoritme Genetika

Algoritme genetika merupakan salah satu bagian dari algoritme evolusi yang populer. Algoritme genetika banyak diterapkan untuk mencari solusi terhadap permasalahan yang kompleks karena algoritme genetika memiliki (Mahmudy et.al., 2013).

Tahap awal dari proses algoritme genetika dimulai dengan inisialisasi yang merupakan proses membentuk individu secara acak sehingga memiliki gen penyusun yaitu kromosom. Kromosom ini mewakili solusi dari permasalahan yang akan dipecahkan. Reproduksi dilakukan untuk menghasilkan keturunan (*offspring*) dari individu-individu yang ada di populasi. Evaluasi digunakan untuk menghitung kebugaran (*fitness*) setiap kromosom. Semakin besar *fitness* maka semakin baik kromosom tersebut untuk dijadikan calon solusi. Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya (Mahmudy, 2013).

2.5.1 Representasi Kromosom

Representasi kromosom digunakan untuk mendefinisikan setiap individu yang ada di dalam suatu populasi (Rennard, 2000). Setiap individu atau kromosom tersusun atas urutan gen dan setiap gen akan memiliki digit representasi kromosom yang diantaranya yaitu representasi *biner* yang merupakan representasi paling umum dan sederhana hanya memiliki nilai 0 dan 1. Representasi permutasi merupakan representasi yang biasa digunakan untuk membentuk solusi pemecahan masalah penjadwalan dan *travel salesmen problem*. Representasi *integer* merupakan representasi yang memiliki nilai bilangan bulat, dan representasi *real code* yang memberikan nilai berupa bilangan *real*.

2.5.2 Inisialisasi Populasi

Inisialisasi berfungsi untuk membentuk suatu himpunan solusi yang baru secara acak sebanyak jumlah kromosom dan diletakkan pada suatu wadah yaitu populasi (*popSize*) yang telah ditentukan besaran jumlah populasinya. Populasi mendefinisikan jumlah individu yang akan dimuat pada suatu populasi (Mahmudy, 2013).

Misalkan dengan menentukan $popSize = 4$ dan menggunakan representasi kromosom bilangan bulat bertipe *integer*. Dapat dilihat pada Tabel 2.2

Tabel 2.3 Contoh Representasi kromosom

Parent	Kromosom				
	x1	x2	x3	x4	x5
P1	5	8	2	1	9
P2	10	3	4	2	7
P3	2	5	6	3	1
P4	7	5	8	9	1

2.5.3 Crossover

Proses *crossover* secara acak melakukan pemilihan dua induk dari populasi. Metode *crossover* yang digunakan adalah metode *extended intermediate crossover*, yang secara acak memilih dua induk dari populasi untuk menghasilkan *offspring* yang dihasilkan sebanyak $cr \times popSize$. Misalkan dengan menggunakan data pada Tabel 2.2 yang terpilih sebagai induk adalah P1 dan P2. Dengan nilai $cr = 0.5$, maka $cr \times popSize = 0.5 \times 4 = 2$ *offspring* yang dihasilkan dari proses *crossover*. Maka akan dihasilkan dua *offspring* (C1 dan C2) sebagai berikut:

$$\begin{aligned}
 C1 &= P1 + \alpha (P2 - P1) \\
 C2 &= P2 + \alpha (P1 - P2)
 \end{aligned}
 \tag{2.3}$$

Secara acak nilai α dibangkitkan pada interval tertentu, contohnya yaitu pada interval $[-1, 1]$. Maka akan menghasilkan *offspring* sebagai berikut:

$$\begin{aligned}
 C1 : x1 &= 5 + 0.5 (10 - 5) = 7.5 \\
 C2 : x2 &= 10 + 0.3 (5 - 10) = 11.5
 \end{aligned}$$

2.5.4 Mutasi

Mutasi dilakukan dengan memilih satu induk secara acak dari populasi. Metode mutasi yang digunakan adalah dengan memilih satu titik acak kemudian mengubah nilai gen pada titik tersebut (Mahmudy, 2013). Misalkan dengan menggunakan data pada Tabel 2.3 yang terpilih sebagai induk adalah P2. Dengan $mr = 0.2$, maka $mr \times popSize = 0.2 \times 4 = 0.8$ (dibulatkan menjadi 1) *offspring* yang dihasilkan dari proses *crossover*. Maka akan dihasilkan *offspring* ketiga (C3) seperti pada Tabel 2.4.

Tabel 2.4 Contoh Reciprocal Exchange Mutation

P2	10	3	4	2	7
C3	10	2	4	3	7

2.5.5 Evaluasi

Evaluasi merupakan proses menghitung nilai *fitness* setiap kromosom. Semakin besar nilai *fitness* yang dimiliki menandakan semakin baik kromosom tersebut untuk menjadi calon solusi.

Pada kasus pencarian nilai maksimum maka nilai *fitness* untuk tiap individu bisa dihitung secara langsung (Mahmudy, 2013) :

$$fitness = f(x)
 \tag{2.4}$$

repository.ub.ac.id

Pada penyelesaian yang berada diluar daerah *visible* dapat digunakan suatu modifikasi fungsi *fitness* dengan menambahkan suatu fungsi yang sering disebut sebagai fungsi *penalty*, sehingga fungsi *fitness* menjadi :

$$fitness = \frac{1}{f(x)+p(x)} \quad (2.5)$$

2.5.6 Seleksi

Melalui proses ini maka terbuatlah generasi baru dimana kromosom diperoleh dari kromosom sebelumnya. Proses seleksi ini digunakan agar hanya kromosom yang baik dapat melanjutkan proses algoritme genetika. Teknik seleksi yang akan digunakan tergantung pada permasalahan yang akan diselesaikan. Ada bermacam-macam teknik seleksi, diantaranya *roulette wheel selection*, *rank base selection*, *stady state selection* dan *elitism selection* (Mahmudy, 2013). Metode seleksi yang digunakan dalam penyelesaian permasalahan penentuan komposisi pakan kuda adalah metode *elitism selection*.

Seleksi *elitism* merupakan metode seleksi yang masing-masing individu pada populasi akan diurutkan secara objektif berdasarkan hasil yang dimiliki. Proses pemilihan individe pada metode *elitism* yaitu dengan memilih kromosom yang memiliki nilai *fitness* tertinggi yang akan bertahan hidup pada generasi selanjutnya sebanyak populasi yang ditentukan. Individu dengan nilai *fitness* yang rendah akan tidak akan bertahan hidup dan hilang saat memasuki generasi yang baru (Suyanto, 2014).



BAB 3 METODOLOGI

Pada bab metodologi membahas tentang metode yang digunakan untuk penelitian yang terdiri beberapa tahapan yaitu studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan pengambilan kesimpulan dalam penelitian ini. Tahapan-tahapan pada penelitian ini akan diilustrasikan dengan diagram alir metodologi penelitian pada Gambar 3.1:



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Metode ini digunakan untuk mengumpulkan referensi dan mendapatkan informasi tambahan dari buku, jurnal, serta skripsi. Mempelajari berbagai literatur yang berhubungan dengan penelitian optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika, diantaranya:

- Kuda
- Kebutuhan nutrisi kuda dewasa
- Kandungan nutrisi jenis-jenis pakan
- Optimasi algoritme genetika

3.2 Pengumpulan Data

Pada penelitian ini dilakukan pengumpulan data dengan cara observasi dan wawancara dengan rincian:

1. Daftar kebutuhan nutrisi pakan kuda dewasa yang diperoleh dari Dr. Wahyu Santoso, M. Sc., Ir. Bambang Edy Santoso, M.P., dan Peternakan Kuda Mega Star Kota Batu.
2. Data dan harga pakan yang didapatkan melalui observasi pada beberapa tempat penjualan pakan ternak di kota Batu pada bulan Februari 2017.

3.3 Analisa Kebutuhan

Analisa kebutuhan dilakukan untuk melakukan identifikasi terkait kebutuhan yang diperlukan untuk melakukan implementasi sistem. Secara keseluruhan, daftar kebutuhan yang diperlukan dalam implementasi sistem yaitu:

1. Kebutuhan perangkat keras
 - Intel (R) Core (TM) i5-5200U CPU @2.20 GHz
 - Memory 4 GB
 - Hard Disk Drive kapasitas 500 GB
 - Layar 14 inch
2. Kebutuhan perangkat lunak
 - Sistem Operasi Windows 10
 - Microsoft Excel 2016
 - NetBeans IDE 8.0
 - Database MySQL
3. Kebutuhan data
 - Data kebutuhan nutrisi kuda
 - Data harga jenis pakan dan kandungan nutrisinya

3.4 Perancangan Sistem

Perancangan dilakukan untuk merancang sistem optimasi komposisi pakan kuda dewasa. Algoritme genetika melakukan pembangkitan kromosom secara acak, kemudian melakukan inialisasi populasi awal, reproduksi individu dengan *crossover* dan mutasi, evaluasi, serta melakukan seleksi. Proses akan terjadi berulang untuk mendapatkan solusi yang terbaik.

3.5 Implementasi Sistem

Implementasi merupakan penerapan hasil perancangan sistem dengan algoritma genetika untuk diaplikasikan pada program menggunakan bahasa pemrograman Java.

3.6 Pengujian dan Analisis

Kinerja algoritme genetika dipengaruhi oleh parameter-parameter yang bersangkutan yaitu ukuran populasi, jumlah generasi, kombinasi *crossover rate*

dan *mutation rate* sehingga perlu dilakukan pengujian terhadap parameter-parameter tersebut. Dari hasil pengujian yang dihasilkan akan dilakukan analisis untuk memberikan kesimpulan.

3.7 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan hingga pengujian dan analisis telah selesai dilakukan. Setelah didapatkan kesimpulan maka tahap saran dilakukan untuk mendefinisikan kesalahan atau kekurangan pada penelitian dan bisa menjadi bahan perbaikan atau pengembangan yang lebih baik pada penelitian berikutnya.



BAB 4 PERANCANGAN

Pada bab perancangan membahas tentang penyelesaian optimasi pakan kuda dewasa menggunakan algoritme genetika, perancangan antarmuka serta perancangan pengujian sistem yang akan digunakan dalam implementasi optimasi komposisi pakan kuda dewasa menggunakan algoritme genetika.

4.1 Formulasi Permasalahan

Permasalahan yang ingin diselesaikan dalam penelitian ini adalah bagaimana menentukan komposisi pakan kuda dewasa dengan mempertimbangkan standar kebutuhan nutrisi yang dibutuhkan seekor kuda dengan biaya yang minimal. Kandungan gizi pada pakan yang dibutuhkan meliputi *Digestible Energy*, protein, kalsium, dan fosfor. Proses penyusunan komposisi pakan kuda dewasa memerlukan data berupa berat badan kuda dan tingkatan kategori kuda yang terbagi menjadi 4 kategori seperti:

- Pemeliharaan
- 90 Hari Sebelum Melahirkan
- Menyusui, 3 Bulan Awal
- Menyusui, 3 Bulan Akhir

Contoh kasus apabila terdapat seekor kuda dewasa dengan berat badan 500 kg dengan tingkat kategori di fase pemeliharaan. Bahan pakan yang tersedia yaitu jerami padi segar, jerami padi kering, dedak gandum, jagung kuning, gaplek, dan onggok. Tahap awal perhitungan yaitu menghitung kebutuhan pakan perhari yang dibutuhkan dengan menggunakan Persamaan (2-1) sebagai berikut:

$$\begin{aligned}\text{Jumlah pakan maksimal kuda perhari} &= \frac{2}{100} \times 500 \\ &= 10 \text{ Kg}\end{aligned}$$

Dari 10 Kg pakan minimum tersebut harus terdiri dari pakan ideal antara serat dan konsentrat seperti pada Persamaan (2-2) sehingga pakan terdiri dari minimal 5 Kg serat dan minimal 2.5 Kg konsentrat.

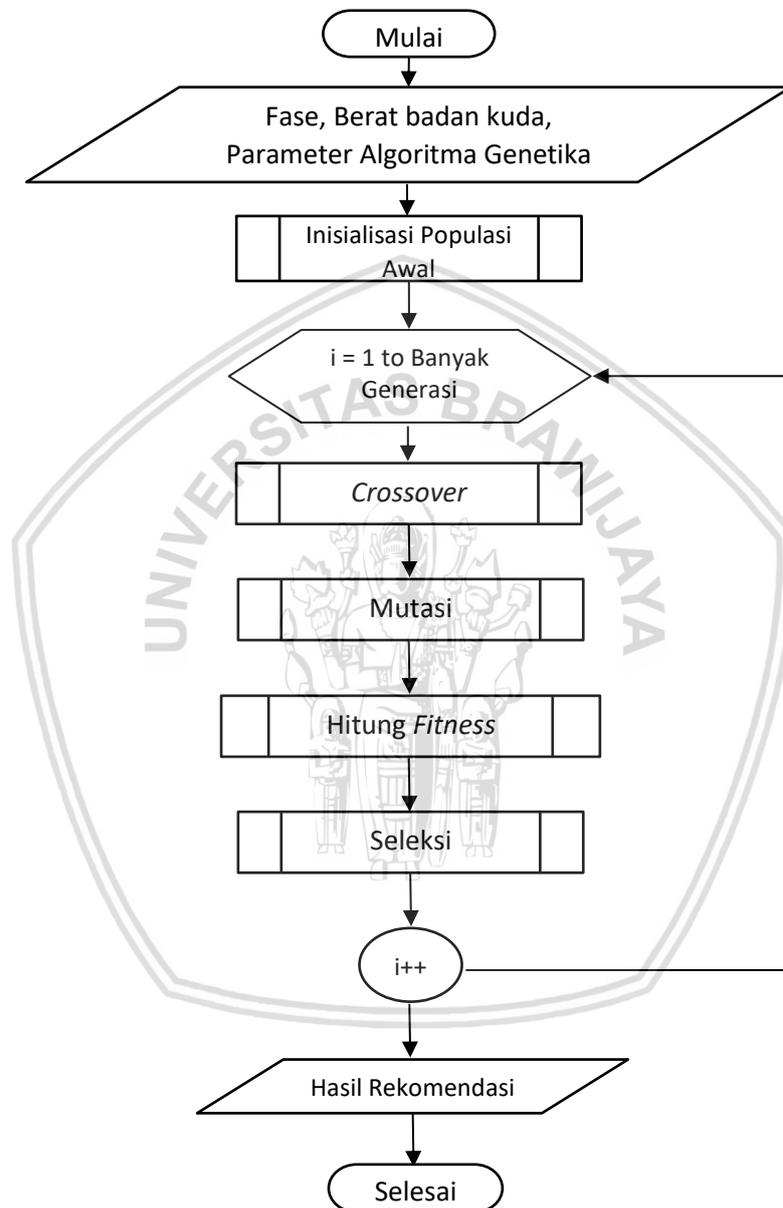
Maka berdasarkan Tabel 2.2 kebutuhan nutrisi yang dibutuhkan adalah sebagai berikut:

- *Digestible Energy* = 16.7 Mcal
- Protein = 630 gr
- Kalsium = 20 gr
- Fosfor = 14 gr

Dengan bahan pakan yang tersedia maka permasalahan yang terjadi adalah berapa komposisi masing-masing bahan pakan yang harus diberikan untuk memenuhi standar nutrisi kuda tersebut dengan biaya yang minimum. Untuk menyelesaikan permasalahan tersebut dapat menggunakan algoritme genetika sebagai solusi.

4.2 Siklus Penyelesaian Masalah Menggunakan Algoritme Genetika

Dalam menyelesaikan permasalahan pada sub bab 4.1 maka digunakan algoritme genetika dengan beberapa langkah yang dapat digambarkan seperti pada Gambar 4.1.



Gambar 4.1 Flowchart Algoritme Genetika

Algoritme genetika digunakan dalam menentukan komposisi pakan kuda dewasa untuk memperoleh hasil yang optimal dengan terpenuhinya standar nutrisi dan harga yang minimum. Untuk parameter algoritme genetika misalkan digunakan data seperti berikut :



- Ukuran populasi : 7
- Jumlah generasi : 1
- Crossover rate : 0.2
- Mutation rate : 0.1

4.2.1 Representasi Kromosom dan Perhitungan Fitness

Representasi kromosom adalah proses pemetaan dari penyelesaian solusi suatu masalah. Suatu kromosom tersusun atas sejumlah gen yang mewakili variabel-variabel keputusan yang digunakan dalam sebuah solusi (Mahmudy, 2013). Pada penelitian ini menggunakan representasi kromosom *real code* dengan menggunakan bilangan *real* yang dibangkitkan secara random pada *interval* [1...10]. Panjang kromosom merupakan banyaknya bahan pakan yang digunakan dalam pemberian pakan yaitu terdiri dari 6 pakan (2 serat dan 4 konsentrat).

Tabel 4.1 Contoh Representasi Kromosom

Individu	Kromosom					
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok
P1	5.0	9.0	6.0	5.0	2.0	3.0

Representasi kromosom pada Tabel 4.1 terdiri atas 6 gen penyusun yaitu jerami padi segar, jerami padi kering, dedak gandum, jagung kuning, gaplek, dan onggok. Contoh perhitungan nilai fitness sebagai berikut:

1. Menghitung bobot bahan pakan

Nilai gen penyusun pada individu dalam Tabel 4.1 merupakan nilai bobot sementara yang tidak memiliki satuan sehingga perlu perhitungan atau normalisasi terhadap bobot gen pada individu ke dalam satuan kilogram agar perhitungan nutrisi dapat sesuai dengan kebutuhan pakan yang diperlukan. Normalisasi bobot pakan pada persamaan (4.1) didapatkan dari penelitian yang telah dilakukan oleh Saraswati (2016).

$$\text{bobot bahan pakan} = \frac{\text{nilai gen}}{\text{total kromosom}} \times \text{total bobot pakan} \tag{4.1}$$

Keterangan:

- Nilai gen = nilai dari setiap gen
- Total kromosom = jumlah setiap gen pada satu individu
- Total bobot pakan = hasil perhitungan jumlah pakan yang dibutuhkan berdasarkan berat kuda menggunakan Persamaan (2.1)

Berdasarkan Persamaan (4.1) maka diperoleh bobot pakan yang telah di normalisasi seperti pada Tabel 4.2.

Tabel 4.2 Perhitungan Bobot Bahan Pakan P1

Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok	Serat	Konsentrat
$\frac{5}{(5+9+6+5+2+3) \times 10}$ $=$ 1.6	$\frac{9}{(5+9+6+5+2+3) \times 10}$ $=$ 3	$\frac{6}{(5+9+6+5+2+3) \times 10}$ $=$ 2	$\frac{5}{(5+9+6+5+2+3) \times 10}$ $=$ 1.6	$\frac{2}{(5+9+6+5+2+3) \times 10}$ $=$ 0.6	$\frac{3}{(5+9+6+5+2+3) \times 10}$ $= 1$	$1.6 + 3 = 4.7$	$2 + 1.6 + 0.6 + 1 = 5.3$

2. Menghitung nilai nutrisi

Kandungan nilai nutrisi (*Digestible Energy*, protein, kalsium, dan fosfor) untuk setiap bahan pakan yang telah dipilih seperti pada Tabel 4.3.

Tabel 4.3 Rincian Data Bahan Pakan

Nama Bahan Pakan	Digestible Energy (Mcal/kg)	Protein (%)	Kalsium (%)	Fosfor (%)
Jerami Padi Segar	1.35	4.3	0	0
Jerami Padi Kering	1.27	3.7	0	0
Dedak Gandum	1.85	11.3	0.06	0.77
Jagung Kuning	3.12	10.3	0.02	0.33
Gaplek	2.6	1.7	0.1	0.04
Onggok	2.45	2.2	0.68	0.05

Karena nilai kandungan nutrisi protein, kalsium, dan fosfor tersebut masih dalam bentuk persen, nilai tersebut harus dibagi dengan 100 agar didapatkan nilai nutrisi dalam satuan Kilogram seperti yang terdapat pada Tabel 4.4.

Tabel 4.4 Kandungan Nutrisi Bahan Pakan

Nama Bahan Pakan	Digestible Energy	Protein	Kalsium	Fosfor
Jerami Padi Segar	1.35	0.043	0	0
Jerami Padi Kering	1.27	0.037	0	0
Dedak Gandum	1.85	0.113	0.0006	0.0077
Jagung Kuning	3.12	0.103	0.0002	0.0033
Gaplek	2.6	0.017	0.001	0.0004
Onggok	2.45	0.022	0.0068	0.0005

Untuk memperoleh nilai nutrisi setiap bahan pakan ditunjukkan pada contoh sebagai berikut:



a. Nilai nutrisi Jerami Padi Segar

$$\begin{aligned} \text{DE} &= \text{Bobot Jerami Padi Segar} \times \text{Nutrisi DE Jerami Padi Segar} \\ &= 1.6 \times 1.35 \\ &= 2.16 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Protein} &= \text{Bobot Jerami Padi Segar} \times \text{Nutrisi Protein Jerami Padi Segar} \\ &= 1.6 \times 0.043 \\ &= 0.0688 \text{ kg} \\ &= 68.8 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Kalsium} &= \text{Bobot Jerami Padi Segar} \times \text{Nutrisi Kalsium Jerami Padi Segar} \\ &= 1.6 \times 0 \\ &= 0 \text{ kg} \\ &= 0 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Fosfor} &= \text{Bobot Jerami Padi Segar} \times \text{Nutrisi Fosfor Jerami Padi Segar} \\ &= 2 \times 0 \\ &= 0 \text{ kg} \\ &= 0 \text{ gr} \end{aligned}$$

b. Nilai nutrisi Jerami Padi Kering

$$\begin{aligned} \text{DE} &= \text{Bobot Jerami Padi Kering} \times \text{Nutrisi DE Jerami Padi Kering} \\ &= 3 \times 1.27 \\ &= 3.81 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Protein} &= \text{Bobot Jerami Padi Kering} \times \text{Nutrisi Protein Jerami Padi Kering} \\ &= 3 \times 0.037 \\ &= 0.111 \text{ kg} \\ &= 111 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Kalsium} &= \text{Bobot Jerami Padi Kering} \times \text{Nutrisi Kalsium Jerami Padi Kering} \\ &= 3 \times 0 \\ &= 0 \text{ kg} \\ &= 0 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Fosfor} &= \text{Bobot Jerami Padi Kering} \times \text{Nutrisi Fosfor Jerami Padi Kering} \\ &= 3 \times 0 \\ &= 0 \text{ kg} \\ &= 0 \text{ gr} \end{aligned}$$

c. Nilai nutrisi Dedak Gandum

$$\begin{aligned} \text{DE} &= \text{Bobot Dedak Gandum} \times \text{Nutrisi DE Dedak Gandum} \\ &= 2 \times 1.85 \\ &= 3.7 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Protein} &= \text{Bobot Dedak Gandum} \times \text{Nutrisi Protein Dedak Gandum} \\ &= 2 \times 0.113 \\ &= 0.226 \text{ kg} \\ &= 226 \text{ gr} \end{aligned}$$

$$\begin{aligned}\text{Kalsium} &= \text{Bobot Dedak Gandum} \times \text{Nutrisi Kalsium Dedak Gandum} \\ &= 2 \times 0.0006 \\ &= 0.0012 \text{ kg} \\ &= 1.2 \text{ gr}\end{aligned}$$

$$\begin{aligned}\text{Fosfor} &= \text{Bobot Dedak Gandum} \times \text{Nutrisi Fosfor Dedak Gandum} \\ &= 2 \times 0.0077 \\ &= 0.0154 \text{ kg} \\ &= 15.4 \text{ gr}\end{aligned}$$

d. Nilai nutrisi Jagung Kuning

$$\begin{aligned}\text{DE} &= \text{Bobot Jagung Kuning} \times \text{Nutrisi DE Jagung Kuning} \\ &= 1.6 \times 3.12 \\ &= 4.992 \text{ kg}\end{aligned}$$

$$\begin{aligned}\text{Protein} &= \text{Bobot Jagung Kuning} \times \text{Nutrisi Protein Jagung Kuning} \\ &= 1.6 \times 0.103 \\ &= 0.1648 \text{ kg} \\ &= 164.8 \text{ gr}\end{aligned}$$

$$\begin{aligned}\text{Kalsium} &= \text{Bobot Jagung Kuning} \times \text{Nutrisi Kalsium Jagung Kuning} \\ &= 1.6 \times 0.0002 \\ &= 0.00032 \text{ kg} \\ &= 0.32 \text{ gr}\end{aligned}$$

$$\begin{aligned}\text{Fosfor} &= \text{Bobot Jagung Kuning} \times \text{Nutrisi Fosfor Jagung Kuning} \\ &= 1.6 \times 0.0033 \\ &= 0.00528 \text{ kg} \\ &= 5.28 \text{ gr}\end{aligned}$$

e. Nilai nutrisi Gaplek

$$\begin{aligned}\text{DE} &= \text{Bobot Gaplek} \times \text{Nutrisi DE Gaplek} \\ &= 0.6 \times 2.6 \\ &= 1.56 \text{ kg}\end{aligned}$$

$$\begin{aligned}\text{Protein} &= \text{Bobot Gaplek} \times \text{Nutrisi Protein Gaplek} \\ &= 0.6 \times 0.017 \\ &= 0.0102 \text{ kg} \\ &= 10.2 \text{ gr}\end{aligned}$$

$$\begin{aligned}\text{Kalsium} &= \text{Bobot Gaplek} \times \text{Nutrisi Kalsium Gaplek} \\ &= 0.6 \times 0.001 \\ &= 0.0006 \text{ kg} \\ &= 6 \text{ gr}\end{aligned}$$

$$\begin{aligned}\text{Fosfor} &= \text{Bobot Gaplek} \times \text{Nutrisi Fosfor Gaplek} \\ &= 0.6 \times 0.0004 \\ &= 0.00024 \text{ kg} \\ &= 0.24 \text{ gr}\end{aligned}$$

f. Nilai nutrisi Onggok

$$\begin{aligned} \text{DE} &= \text{Bobot Onggok} \times \text{Nutrisi DE Onggok} \\ &= 1 \times 2.45 \\ &= 2.45 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Protein} &= \text{Bobot Onggok} \times \text{Nutrisi Protein Onggok} \\ &= 1 \times 0.022 \\ &= 0.022 \text{ kg} \\ &= 22 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Kalsium} &= \text{Bobot Onggok} \times \text{Nutrisi Kalsium Onggok} \\ &= 1 \times 0.0068 \\ &= 0.0068 \text{ kg} \\ &= 6.8 \text{ gr} \end{aligned}$$

$$\begin{aligned} \text{Fosfor} &= \text{Bobot Onggok} \times \text{Nutrisi Fosfor Onggok} \\ &= 1 \times 0.0005 \\ &= 0.0005 \text{ kg} \\ &= 0.5 \text{ gr} \end{aligned}$$

3. Menghitung nilai *penalty*

Setelah mendapatkan nilai kandungan nutrisi (DE, Protein, Kalsium, Fosfor) maka langkah selanjutnya yaitu menjumlahkan masing-masing nilai nutrisinya.

$$\begin{aligned} - \text{ DE} &= 2.16 + 3.81 + 3.7 + 4.9 + 1.56 + 2.45 &= 19.1 \\ - \text{ Protein} &= 68.8 + 111 + 226 + 164.8 + 10.2 + 22 &= 613.7 \\ - \text{ Kalsium} &= 0 + 0 + 1.2 + 0.32 + 0.32 + 6 + 6.8 &= 9 \\ - \text{ Fosfor} &= 0 + 0 + 15.4 + 5.28 + 0.24 + 0.5 &= 21.7 \end{aligned}$$

Sehingga didapatkan total nutrisi yang tersedia dari pakan individu P1, maka dapat dibuat perbandingan antara kebutuhan nutrisi dengan nutrisi yang tersedia seperti pada Tabel 4.5.

Tabel 4.5 Kebutuhan dan Ketersediaan Nutrisi

	DE	Protein	Kalsium	Fosfor	Serat	Konsentrat
Kebutuhan	16.7	630	20	14	5	2.5
Tersedia	19.1	613.7	9.0	21.7	4.7	5.3
<i>Penalty</i>	0	16.3	11	0	0.3	0

Berdasarkan Tabel 4.5, total dari *penalty* dapat diperoleh dengan menjumlahkan nilai *penalty* tiap gen sehingga total *penalty* adalah 27.6 Nilai *penalty* merupakan nilai pelanggaran yang terjadi akibat adanya nutrisi yang tidak terpenuhi sesuai standar berdasarkan nilai ketersediaan nutrisi yang lebih kecil dari nilai kebutuhan nutrisi. Nilai *penalty* bernilai = 0 apabila nilai tersedia lebih dari atau sama dengan nilai kebutuhan. Apabila nilai nutrisi tersedia lebih kecil dari nilai nutrisi kebutuhan maka digunakan Persamaan (4.2).

$$Penalty = \begin{cases} 0, & TerNut \geq KebNut \\ KebNut - TerNut, & TerNut < KebNut \end{cases} \quad (4.2)$$

4. Menghitung harga

Karena perbedaan harga bahan pakan yang ada pada setiap tempat, pada penelitian ini digunakan harga bahan pakan yang diperoleh dari hasil survey pada kota Batu. Pada proses optimasi komposisi pakan, harga setiap jenis bahan pakan didapatkan dengan cara mengalikan bobot bahan pakan yang digunakan dengan harga seperti pada Persamaan (4.3).

$$Total\ Harga = \text{bobot bahan pakan} \times \text{harga} \quad (4.3)$$

Dari bobot bahan pakan yang didapatkan pada Tabel 4.2 maka harga untuk setiap bahan pakan dapat dihitung dengan menggunakan daftar harga bahan pakan pada Lampiran A. Sehingga diperoleh harga untuk setiap bahan pakan sebagai berikut:

- Jerami Padi Segar = 1.6 x 5000 = 8000
- Jerami Padi Kering = 3 x 6000 = 18000
- Dedak Gandum = 2 x 4500 = 9000
- Jagung Kuning = 2 x 6000 = 12000
- Gaplek = 1.6 x 6000 = 9600
- Onggok = 1 x 2000 = 2000

Setelah didapatkan harga dari masing masing bahan pakan, maka diperoleh total harga seperti pada Tabel 4.6.

Tabel 4.6 Total Harga Individu P1

Individu	Kromosom						Harga
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok	
P1	5.0	9.0	6.0	5.0	2	3	49000

5. Menghitung Fitness

Setelah memperoleh total harga, langkah selanjutnya yaitu menghitung nilai fitness setiap individu. Nilai *fitness* berfungsi untuk mengetahui baik tidaknya suatu individu untuk menjadi solusi (Mahmudy, 2013).

Pada penelitian ini, solusi yang ingin dibangun adalah memenuhi standar minimal kebutuhan nutrisi dengan biaya yang minimum sehingga dicari harga dari total pakan yang minimum dan penalty yang kecil. Maka perhitungan untuk mendapatkan nilai fitness dengan menggunakan Persamaan (4.4).

$$fitness = \frac{10000}{\text{Harga} + (\text{penalty} \times C)} \quad (4.4)$$

Keterangan:
 Harga = total harga

Penalty = total *penalty*
C = konstanta yang digunakan untuk menyeimbangkan *penalty* dengan harga (10000)

Nilai konstanta ditetapkan 10000 agar jika dikali dengan nilai *penalty* dapat seimbang dengan nilai harga bahan pakan yang berkisar antara ribuan dan puluhan ribu dan juga agar menjaga setiap individu yang memiliki harga minimum namun memiliki nilai *penalty* tetap memiliki nilai *fitness* yang jauh lebih kecil dari pada individu yang memiliki harga relatif lebih mahal tetapi tidak memiliki nilai *penalty*.

Hasil perhitungan nilai *fitness* dari individu P1 dapat dilihat pada Tabel 4.7.

Tabel 4.7 Nilai *fitness* P1

Individu	Kromosom						Harga	<i>Penalty</i>	<i>Fitness</i>
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok			
P1	5.0	9.0	6.0	5.0	2.0	3.0	49000	30.5	0.028

Nilai *fitness* diperoleh dengan perhitungan sebagai berikut:

$$fitness = \frac{10000}{49000 + (30.5 \times 10000)} = 0.028$$

4.2.2 Inisialisasi Populasi Awal

Pada tahap awal, pembangkitan populasi awal diperoleh sebanyak jumlah *popSize* yang telah ditentukan dengan panjang kromosom berjumlah sesuai dengan jumlah bahan pakan yang digunakan. Proses pembangkitan populasi awal dapat dilihat pada Gambar 4.2.

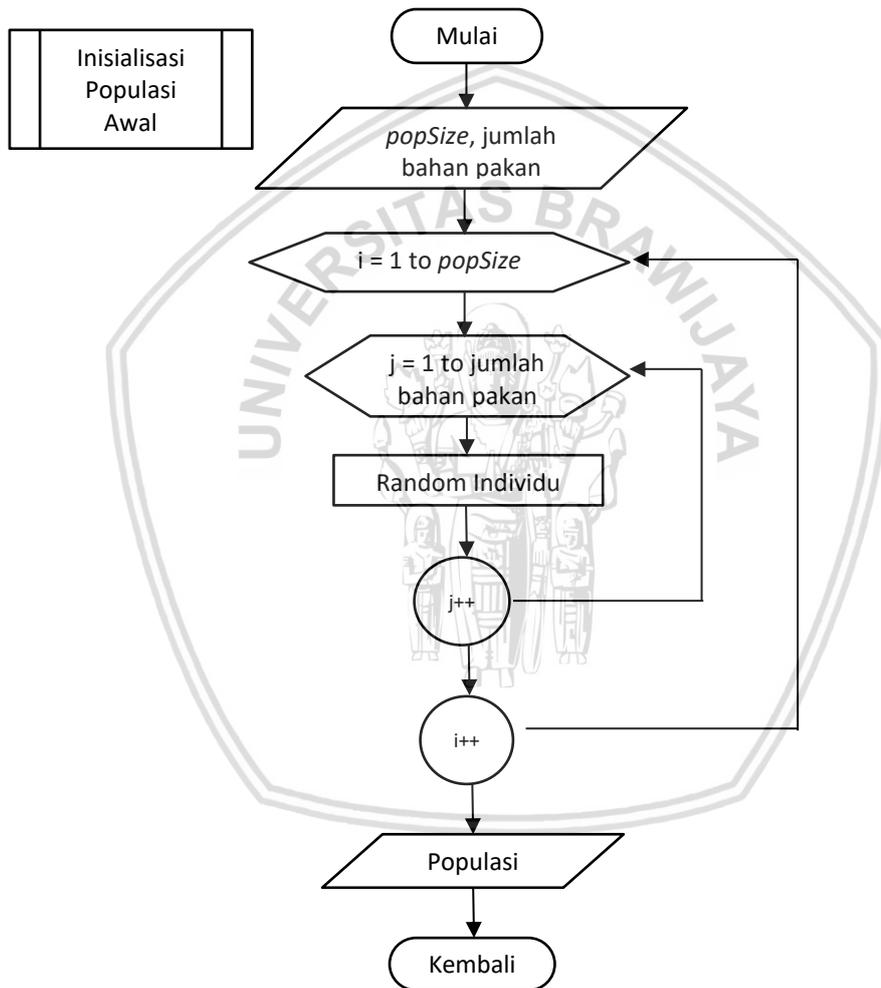
Pada langkah pertama pembangkitan populasi awal adalah dengan memasukkan nilai *popSize*. Kemudian sistem akan menentukan nilai gen dengan range yang telah ditentukan. Setelah didapatkan nilai gen untuk suatu individu, akan dilakukan perulangan sebanyak ukuran populasi yang telah ditentukan.

Pada contoh perhitungan ini menggunakan *popSize* sebanyak 7. Dengan jumlah 6 bahan pakan yang digunakan sehingga setiap individu memiliki panjang kromosom sebesar 6. Dengan menggunakan representasi *real code* dan nilai *range* masing-masing gen dibuat *random* pada *interval* [1;10]. Populasi awal yang telah dibangkitkan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Pembangkitan Populasi Awal

Individu	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok
P1	5.0	9.0	6.0	5.0	2.0	3.0

Individu	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok
P2	8.0	6.0	5.0	5.0	4.0	9.0
P3	9.0	10.0	2.0	4.0	5.0	3.0
P4	9.0	2.0	6.0	7.0	6.0	8.0
P5	4.0	5.0	6.0	8.0	7.0	5.0
P6	3.0	1.0	8.0	2.0	9.0	1.0
P7	4.0	2.0	6.0	2.0	10.0	1.0



Gambar 4.2 Flowchart Pembangkitan Populasi Awal

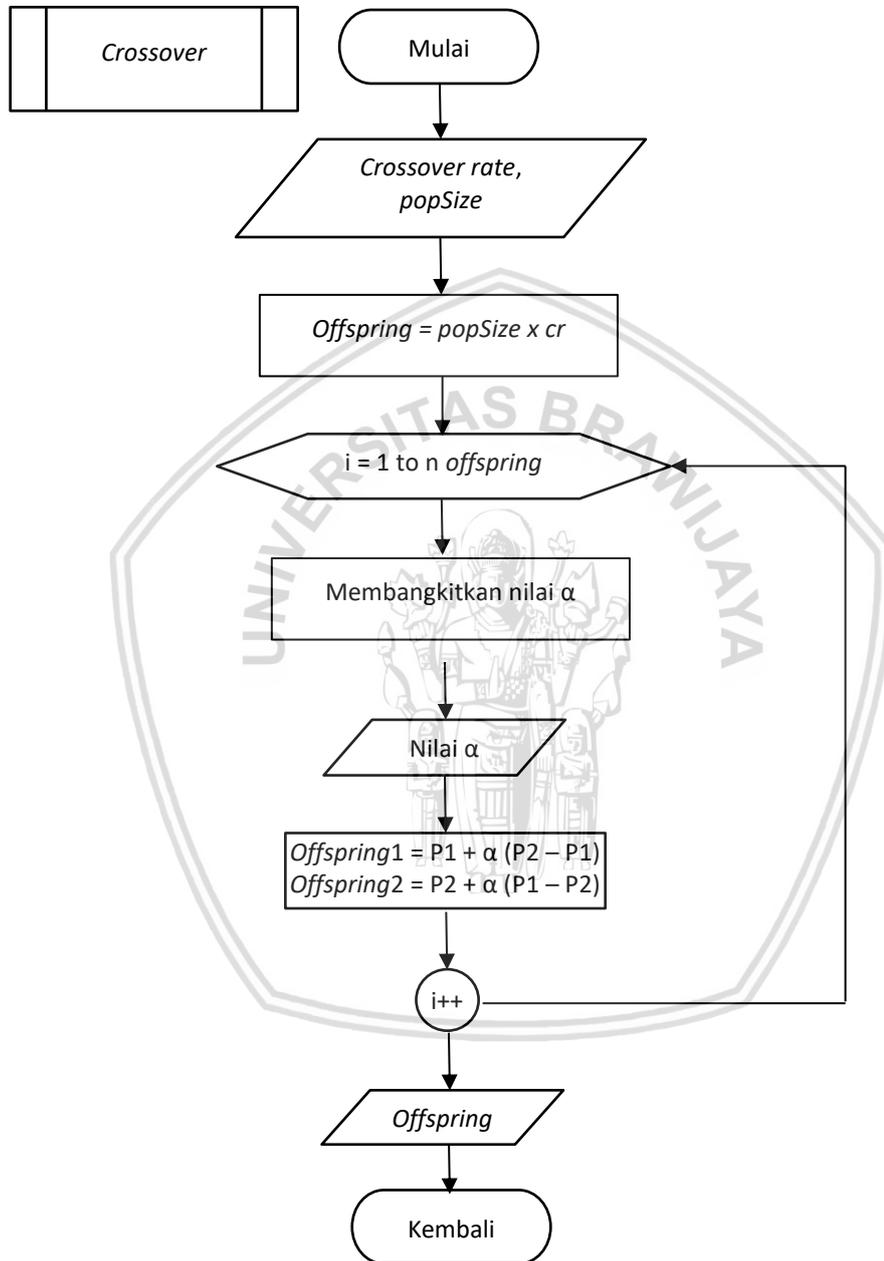
4.2.3 Reproduksi

Terdapat 2 jenis reproduksi yang ada pada algoritme genetika yaitu *crossover* dan *mutation* (Mahmudy, 2013). Pada penelitian ini menggunakan proses reproduksi dalam menggunakan metode *extended intermediate crossover* dan *reciprocal exchange mutation*.



4.2.3.1 Crossover

Metode *crossover* yang digunakan dalam penelitian ini adalah *extended intermediate* untuk menghasilkan *offspring* dari kombinasi dua induk yang terpilih secara random. Proses perhitungan *crossover extended intermediate* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Flowchart Proses Crossover

Berdasarkan *flowchart* pada Gambar 4.3, proses *crossover* diawali dengan menghitung jumlah *offspring* yang akan dihasilkan dengan mengalikan nilai *crossover rate* (*cr*) dengan jumlah populasi (*popSize*).kemudian memilih induk sebanyak jumlah *offspring* dan membangkitkan nilai α pada range yang telah



ditentukan. Dengan menggunakan persamaan (4.5) dan (4.6) maka didapatkan nilai *offspring* 1 dan *offspring* 2.

$$\text{offspring 1} = P1 \times \alpha (P2 - P1) \tag{4.5}$$

$$\text{offspring 2} = P2 \times \alpha (P1 - P2) \tag{4.6}$$

Keterangan:

P1, P2 = Parent1, Parent2

α = nilai random interval (-1;1)

Berdasarkan contoh kasus dengan ketentuan nilai *crossover rate* = 0.2 dan *popSize* = 7 maka *offspring* yang dihasilkan yaitu $0.2 \times 7 = 2$. Kemudian memilih 2 induk secara *random* yaitu P1 dan P2 untuk dilakukan *crossover* dan membangkitkan nilai α kemudian menggunakan persamaan (4.5) dan (4.6) untuk mendapatkan *offspring* seperti yang dapat dilihat pada Tabel 4.9, Tabel 4.10, dan Tabel 4.11.

Tabel 4.9 Parent Crossover Terpilih

P1	5.0	9.0	6.0	5.0	2.0	3.0
P2	8.0	6.0	5.0	5.0	4.0	9.0

Tabel 4.10 Membangkitkan Nilai α

α	0.25	0.25	0.5	0.4	-0.2	-0.1
----------	------	------	-----	-----	------	------

Tabel 4.11 Hasil *Offspring Crossover*

C1	5.75	8.25	5.5	5.0	1.6	2.4
C2	7.25	8.75	8.5	8.0	7.6	7.7

Hasil perhitungan *crossover* diatas dapat dijelaskan sebagai contoh berikut:

$$\begin{aligned}
 C1, \text{ gen 1} &= 5 + (0.25 \times (8-5)) = 5.75 & C2, \text{ gen 1} &= 8 + (0.25 \times (5-8)) = 7.25 \\
 C1, \text{ gen 2} &= 9 + (0.25 \times (6-9)) = 8.25 & C2, \text{ gen 2} &= 6 + (0.25 \times (9-6)) = 8.75 \\
 C1, \text{ gen 3} &= 6 + (0.5 \times (5-6)) = 5.5 & C2, \text{ gen 3} &= 5 + (0.5 \times (6-5)) = 8.5 \\
 C1, \text{ gen 4} &= 5 + (0.4 \times (5-5)) = 5.0 & C2, \text{ gen 4} &= 5 + (0.4 \times (5-5)) = 8.0 \\
 C1, \text{ gen 5} &= 2 + (-0.2 \times (4-2)) = 1.6 & C2, \text{ gen 5} &= 4 + (-0.2 \times (2-4)) = 7.6 \\
 C1, \text{ gen 6} &= 3 + (-0.1 \times (9-3)) = 2.4 & C2, \text{ gen 6} &= 9 + (-0.1 \times (3-9)) = 7.7
 \end{aligned}$$

4.2.3.2 Mutasi

Metode mutasi yang digunakan adalah *reciprocal exchange mutation* untuk menghasilkan *offspring* dari individu yang dipilih secara *random*. Proses perhitungan mutasi *reciprocal exchange mutation* dapat dilihat pada Gambar 4.4.

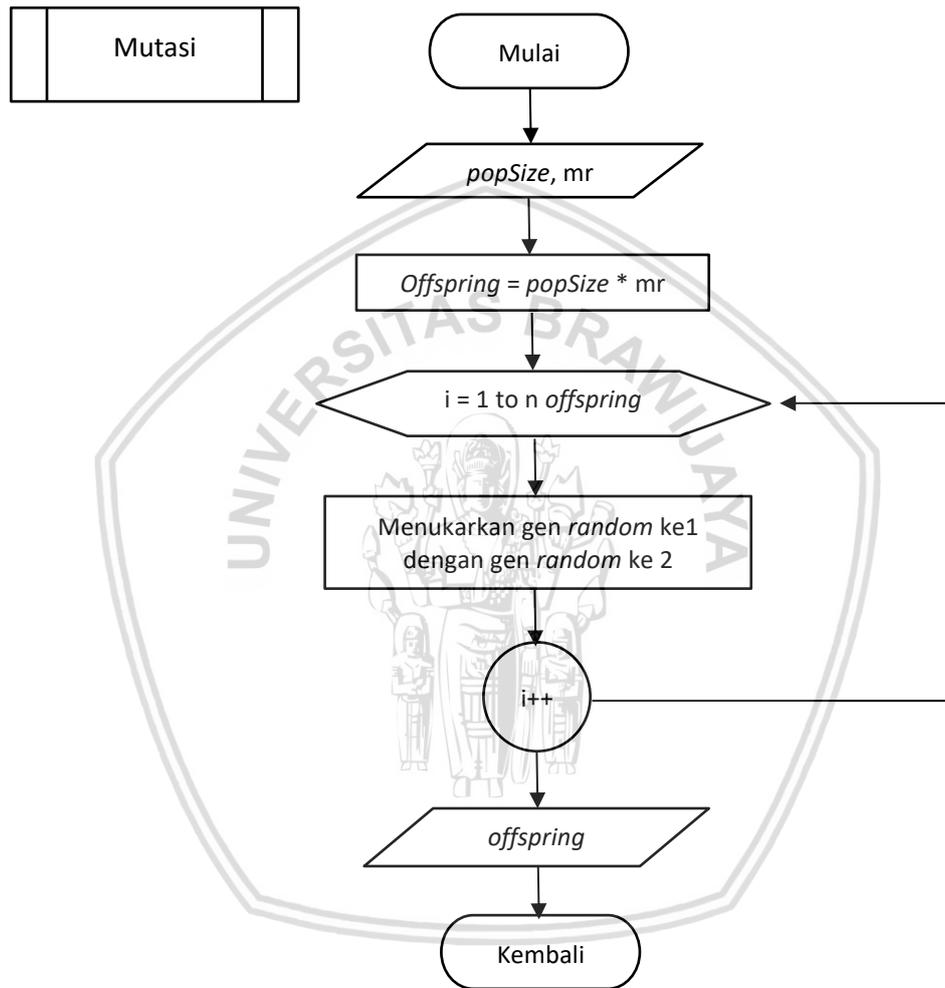
Dari contoh kasus diperoleh *offspring* mutasi = $0.1 \times 7 = 1$. Pemilihan *parent* secara *random* untuk proses mutasi kemudian dipilih 2 gen secara *random* yang akan menjalankan proses mutasi dengan bertukar informasi seperti pada Tabel 4.12 dan Tabel 4.13.

Tabel 4.12 Parent Mutasi Terpilih

P3	9.0	10.0	2.0	4.0	5.0	3.0
----	-----	------	-----	-----	-----	-----

Tabel 4.13 Hasil Offspring Mutasi

C3	9.0	4.0	2.0	10.0	5.0	3.0
----	-----	-----	-----	------	-----	-----



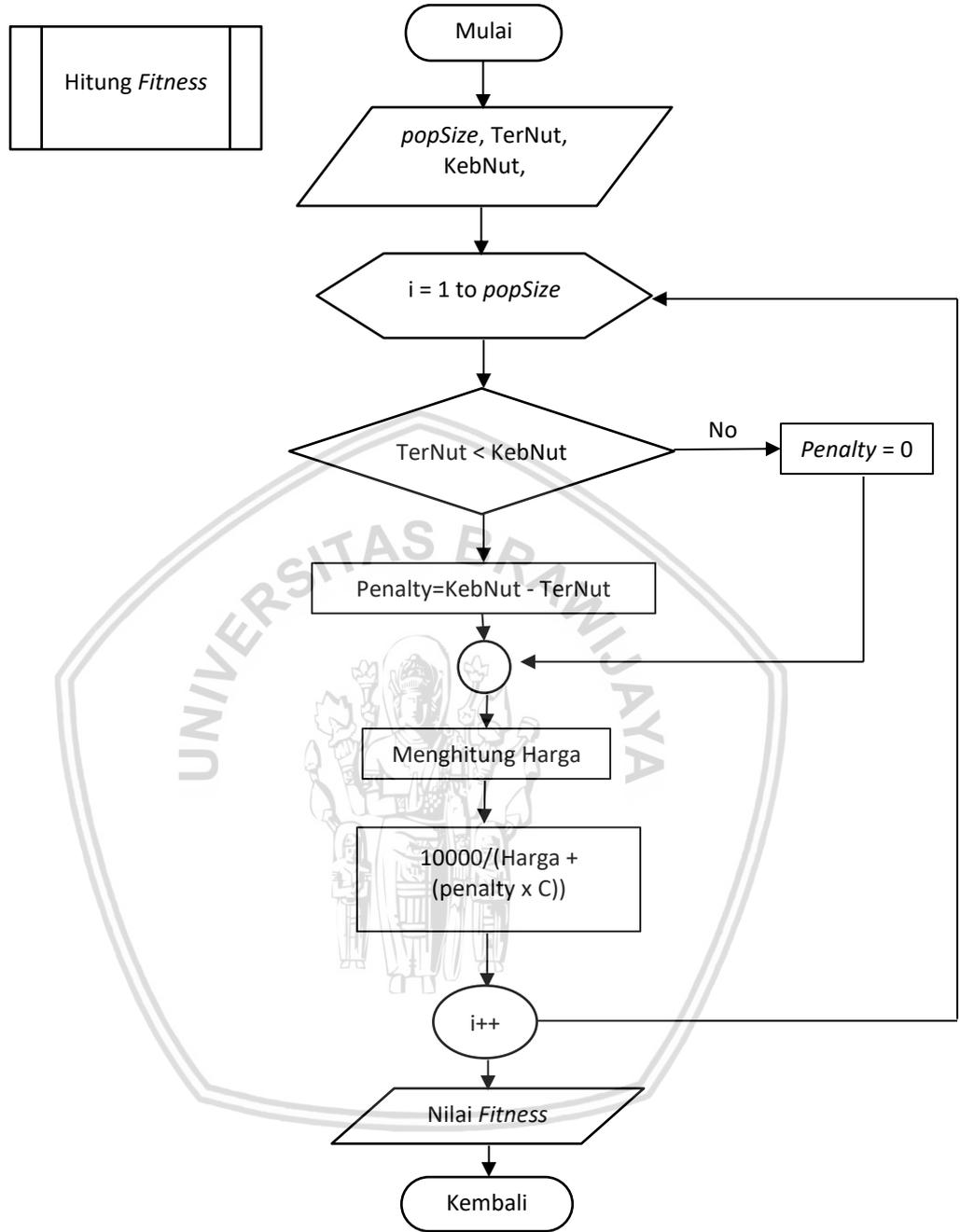
Gambar 4.4 Flowchart Proses Mutasi

4.2.4 Evaluasi

Proses evaluasi dilakukan setelah proses reproduksi bertujuan untuk menghitung nilai *fitness* dari setiap kromosom. Proses perhitungan nilai *fitness* dapat dilihat pada Gambar 4.5.

Hasil perhitungan nilai *fitness* pada seluruh individu pada Tabel 4.14.





Gambar 4.5 Flowchart Proses Hitung Fitness

Pada perhitungan nilai fitness menggunakan konstanta 10000 untuk menyesuaikan harga bahan pakan yang berkisar ribuan hingga puluhan ribu.

Tabel 4.14 Hasil Evaluasi

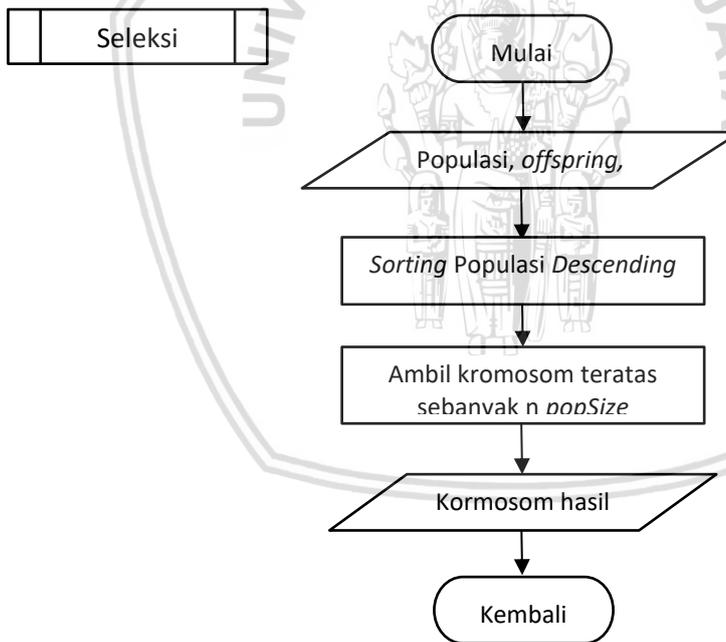
	Kromosom						Harga	Penalty	fitness
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok			
P1	5.0	9.0	6.0	5.0	2.0	3.0	49000	30.5	0.028248588



	Kromosom						Harga	Penalty	fitness
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok			
p2	8.0	6.0	5.0	5.0	4.0	9.0	42297.2973	119.472973	0.008083898
p3	9.0	10.0	2.0	4.0	5.0	3.0	47424.24242	179.2272727	0.005435678
p4	9.0	2.0	6.0	7.0	6.0	8.0	41315.78947	76.86842105	0.012345679
p5	4.0	5.0	6.0	8.0	7.0	5.0	43571.42857	47.58571429	0.019251925
p6	3.0	1.0	8.0	2.0	9.0	1.0	38958.33333	45.83333333	0.020108923
p7	4.0	2.0	6.0	2.0	10.0	1.0	39200	120.58	0.008032129
C1	5.0	7.8	5.6	5.0	2.4	3.0	48263.88901	33.69444693	0.025959977
C2	8.0	7.2	5.4	5.0	3.6	9.0	43062.82709	114.2696323	0.008433416
C3	9.0	3.0	2.0	4.0	5.0	10.0	38939.39394	201.7727273	0.004862237

4.2.5 Seleksi

Metode seleksi yang digunakan adalah elitism selection untuk menentukan individu yang akan bertahan dan digunakan dalam generasi selanjutnya. Proses seleksi dengan metode *elitism selection* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Flowchart Proses Seleksi

Proses seleksi dilakukan untuk mendapatkan individu baru dari hasil proses algoritme genetika untuk generasi yang baru dengan memilih individu yang memiliki nilai *fitness* terbesar sebanyak *popSize* yang telah ditentukan diawal. Berdasarkan contoh kasus, hasil seleksi dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Seleksi

	Kromosom						Harga	Penalty	fitness
	Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Gaplek	Onggok			
P1	5.0	9.0	6.0	5.0	2.0	3.0	49000	30.5	0.028248588
C1	5.0	7.8	5.6	5.0	2.4	3.0	48263.88901	33.69444693	0.025959977
P6	3.0	1.0	8.0	2.0	9.0	1.0	38958.33333	45.83333333	0.020108923
P5	4.0	5.0	6.0	8.0	7.0	5.0	43571.42857	47.58571429	0.019251925
P4	9.0	2.0	6.0	7.0	6.0	8.0	41315.78947	76.86842105	0.012345679
C2	8.0	7.2	5.4	5.0	3.6	9.0	43062.82709	114.2696323	0.008433416
P2	8.0	6.0	5.0	5.0	4.0	9.0	42297.2973	119.472973	0.008083898

Hasil dari seleksi tersebut menunjukkan bahwa P1 merupakan hasil optimasi terbaik dengan nilai *fitness* 0.028248588. ketujuh individu diatas merupakan individu terpilih yang bisa melanjutkan proses di generasi berikutnya.

4.3 Perancangan Antarmuka

Pada perancangan antarmuka aplikasi optimasi pakan kuda dewasa terdapat 2 halaman antarmuka yaitu halaman input data kuda dan bahan pakan dan halaman algoritme genetika.

4.3.1 Rancangan Tampilan Halaman *Input Data Kuda dan Bahan Pakan*

Optimasi Komposisi Pakan Kuda Dewasa Dengan Algoritma Genetika

Input Data

Jenis Kuda 1

Berat Kuda

2 OK

Kebutuhan Nutrisi Kuda

3

Pilih Bahan Pakan

4

5 OK

Pakan yang dipilih

6

7 Optimasi

Gambar 4.7 Rancangan Tampilan Halaman *Input Data Kuda dan Bahan Pakan*

Keterangan Gambar 4.7 :

1. Tempat memasukkan data kuda seperti jenis kuda dan berat kuda.

2. Tombol untuk menampilkan kebutuhan nutrisi kuda.
3. Tempat kebutuhan nutrisi kuda berdasarkan data yang dimasukkan.
4. Tempat memilih jenis pakan kuda.
5. Tombol untuk menampilkan kandungan nutrisi pakan yang dipilih.
6. Tempat menampilkan kandungan nutrisi pakan yang telah terpilih.
7. Tombol untuk menuju halaman algoritme genetika.

4.3.2 Rancangan Tampilan Halaman Algoritme Genetika

Optimasi Komposisi Pakan Kuda Dewasa Dengan Algoritma Genetika

Input Parameter Algoritma Genetika	Hasil Rekomendasi
<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="display: flex; justify-content: space-between;"> Jumlah Populasi <input style="width: 100px;" type="text"/> </div> <div style="display: flex; justify-content: space-between;"> Jumlah Generasi <input style="width: 100px;" type="text"/> </div> <div style="display: flex; justify-content: space-between;"> Crossover rate <input style="width: 100px;" type="text"/> </div> <div style="display: flex; justify-content: space-between;"> Mutation rate <input style="width: 100px;" type="text"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> 1 2 </div> <div style="text-align: right; margin-top: 5px;"> <input type="button" value="Proses"/> </div>	<div style="border: 1px solid black; width: 100px; height: 100px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> 3 </div>
<p>Populasi</p> <div style="border: 1px solid black; width: 100%; height: 60px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; align-items: center;"> 4 </div>	
<div style="display: flex; justify-content: space-between; align-items: center;"> Back 5 </div>	

Gambar 4.8 Rancangan Tampilan Halaman Algoritme Genetika

Keterangan Gambar 4.8 :

1. Tempat untuk memasukkan parameter algoritme genetika yaitu jumlah populasi, nilai *crossover rate* dan *mutation rate*, serta jumlah generasi.
2. Tombol untuk melakukan proses perhitungan algoritme genetika.
3. Tempat untuk menampilkan hasil rekomendasi sistem.
4. Tempat untuk menampilkan populasi.
5. Tombol untuk kembali ke halaman sebelumnya.

4.4 Perancangan Pengujian Sistem

Untuk melakukan evaluasi pada sistem yang akan dibangun pada penelitian ini dilakukan beberapa pengujian agar sistem mampu menghasilkan solusi terbaik dan optimal. Uji coba tersebut antara lain:

1. Pengujian untuk menentukan ukuran populasi yang terbaik.
2. Pengujian untuk menentukan ukuran generasi yang terbaik.



3. Pengujian untuk mencari kombinasi *crossover rate* (cr) dan *mutation rate* (mr) yang terbaik.

4.4.1 Pengujian Nilai Populasi

Pengujian nilai populasi adalah pengujian yang bertujuan untuk dapat mengetahui jumlah atau ukuran populasi yang tepat sehingga mampu memberikan solusi yang optimal dalam suatu masalah, termasuk kasus untuk optimasi komposisi pakan kuda dewasa. Ukuran populasi yang digunakan pada kasus ini adalah kelipatan 10 dimulai dari 50 sampai dengan 140 sebanyak 10 kali percobaan. Rancangan uji coba dalam menentukan ukuran populasi dapat dilihat pada Tabel 4.16.

Tabel 4.16 Rancangan Pengujian Populasi

Ukuran Populasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan ke-										
	1	2	3	4	5	6	7	8	9	10	
50											
60											
70											
80											
90											
100											
110											
120											
130											
140											

4.4.2 Pengujian Nilai Generasi

Pengujian nilai generasi adalah pengujian yang bertujuan untuk mengetahui jumlah atau ukuran generasi yang tepat sehingga mampu memberikan solusi yang optimal dalam menyelesaikan suatu masalah. Ukuran generasi yang digunakan serta untuk diujikan pada kasus ini adalah kelipatan 50 dimulai dari 50 sampai 400 sebanyak 10 kali percobaan. Rancangan uji coba dalam menentukan ukuran populasi dapat dilihat pada Tabel 4.17.

Tabel 4.17 Rancangan Pengujian Generasi

Ukuran Generasi	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>
	Percobaan ke-										
	1	2	3	4	5	6	7	8	9	10	
50											



100												
150												
200												
250												
300												
350												
400												

4.4.3 Pengujian Kombinasi *Crossover Rate* dan *Mutation Rate*

Pengujian kombinasi *crossover rate* dan *mutation rate* adalah pengujian yang bertujuan untuk mengetahui jumlah atau ukuran kombinasi *crossover rate* dan *mutation rate* yang tepat sehingga mampu memberikan solusi yang optimal dalam menyelesaikan suatu masalah. Terdapat 10 kombinasi *crossover rate* dan *mutation rate* yang digunakan serta diujikan pada kasus ini adalah sebanyak 10 kali. Rancangan uji coba dalam menentukan ukuran populasi dapat dilihat pada Tabel 4.18.

Tabel 4.18 Rancangan Pengujian Kombinasi *crossover rate* dan *mutation rate*

CR	MR	Nilai <i>fitness</i>										Rata-rata <i>fitness</i>	
		Percobaan ke-											
		1	2	3	4	5	6	7	8	9	10		
1	0												
0.9	0.1												
0.8	0.2												
0.7	0.3												
0.6	0.4												
0.5	0.5												
0.4	0.6												
0.3	0.7												
0.2	0.8												
0.1	0.9												
0	1												

4.4.4 Pengujian Validasi dengan Parameter Terbaik

Pengujian validasi dengan parameter terbaik adalah pengujian yang bertujuan untuk membandingkan hasil rekomendasi komposisi bahan pakan yang diberikan sistem menggunakan parameter algoritme genetika terbaik yang telah didapatkan



dengan pemberian komposisi pakan yang selama ini dilakukan oleh peternak. Terdapat 4 kombinasi komposisi pakan yang akan dilakukan pengujian validasi sebanyak 3 kali. Rancangan uji coba validasi dapat dilihat pada Tabel 4.19.

Tabel 4.19 Rancangan Pengujian Validasi dengan Parameter Terbaik

Pakan 1	Pakan 2	Pakan 3	Pakan 4	Pakan 5	Pakan 6	Harga	Penalty	Fitness	Harga Awal



BAB 5 IMPLEMENTASI

Pada bab implementasi ini akan dibahas tentang implementasi sistem yang telah dibuat. Pembahasan implementasi meliputi implementasi program dan implementasi antarmuka.

5.1 Implementasi Program

Pada sub bab ini akan dijelaskan tentang implementasi program optimasi pakan kuda dengan algoritme genetika berdasarkan perancangan sistem yang sudah dijelaskan pada bab 4.

5.1.1 Implementasi Inisialisasi Populasi Awal

Pada proses ini digunakan untuk inisialisasi kromosom dengan *random*, panjangnya kromosom sesuai dengan banyaknya bahan pakan yang digunakan. Proses inisialisasi populasi awal dapat dilihat pada *Source Code 5.1*.

```

1 private void inisialisasi() {
2     inisialvariabel();
3     kromosom_awal = new double[getPopSize()][6];
4     System.out.println("Inisialisasi Awal");
5     for (int i = 0; i < getPopSize(); i++) {
6         for (int j = 0; j < 6; j++) {
7             kromosom_awal[i][j]=(random.nextInt(10)+1);
8             System.out.print(kromosom_awal[i][j] + " ");
9         }
10        System.out.println();
11    }

```

Source Code 5.1 Inisialisasi Populasi Awal

Penjelasan *source code* inisialisasi populasi awal adalah sebagai berikut:

1. Baris 3 membuat variabel *array* 2 dimensi untuk menyimpan jumlah kromosom.
2. Baris 5-9 merupakan proses perulangan untuk membentuk kromosom secara acak.

5.1.2 Implementasi Proses *Crossover*

Metode *crossover* yang digunakan adalah *extended intermediate crossover*. Untuk mendapatkan nilai *crossover* terlebih dahulu dilakukan pemilihan individu untuk dilakukan *crossover*. Untuk mendapatkan nilai *crossover* dilakukan juga pembangkitan nilai acak pada tiap kromosomnya. Proses *crossover* ini dilakukan untuk mendapatkan *offspring* atau individu baru, jumlah *offspring* dari *crossover* ini dihasilkan dari perkalian *crossover rate* dengan jumlah populasinya. Proses *crossover* dapat dilihat pada *Source Code 5.2*.

```

1 OffspringC = new double[offspringCrossover][6];
2 double minAlpha = -1;
3 double maxAlpha = 1;
4 double nilaiAlpha[] = new double[6];
5 System.out.println("Nilai alpha :");

```

```

6   for (int i = 0; i < 6; i++) {
7       double acak2 = random.nextInt(10);
8       nilaiAlpha[i] = minAlpha+acak2 / 10 * (maxAlpha - minAlpha);
9       System.out.println(nilaiAlpha[i]);
10  }
11  int last = random_cr.length - 1;
12  int start = 0;
13  while (start < last) {
14      int kromosom_atas;
15      int kromosom_bawah;
16      kromosom_atas = random_cr[start] - 1;
17      kromosom_bawah = random_cr[last] - 1;
18      for (int i = 0; i < nilaiAlpha.length; i++) {
19  OffspringC[start][i] = kromosom_awal[kromosom_atas][i] +
20  nilaiAlpha[i] * (kromosom_awal[kromosom_bawah][i] -
21  kromosom_awal[kromosom_atas][i]);
22  OffspringC[last][i] = kromosom_awal[kromosom_bawah][i] +
23  nilaiAlpha[i] * (kromosom_awal[kromosom_atas][i] -
24  kromosom_awal[kromosom_bawah][i]);
25  }
26  last--;
27  start++;
28  }
29  System.out.println("Hasil Crossover");
30  for (int i = 0; i < OffspringC.length; i++) {
31  for (int j = 0; j < nilaiAlpha.length; j++) {
32  System.out.print(OffspringC[i][j] + " ");
33  }
34  System.out.println();
35  }

```

Source Code 5.2 Proses Crossover

Penjelasan *source code* proses *crossover* adalah sebagai berikut:

1. Baris 3 membuat variabel *array* 2 dimensi untuk menyimpan jumlah *offspring*.
2. Baris 6-10 merupakan proses perulangan untuk membentuk nilai alpha secara acak.
3. Baris 11-35 merupakan proses perhitungan *crossover* dengan nilai alpha yang telah didapatkan.

5.1.3 Implementasi Proses Mutasi

Metode mutasi yang digunakan adalah *reciprocal exchange mutation*. Untuk mendapatkan nilai mutasi terlebih dahulu dilakukan pemilihan individu untuk dilakukan mutasi. Kromosom dipilih secara acak sebanyak dua untuk dilakukan pertukaran. Kromosom yang tidak terpilih maka nilai kromosomnya tetap. Proses mutasi ini dilakukan untuk menghasilkan *offspring*, jumlah *offspring* yang dihasilkan adalah sejumlah hasil kali *mutation rate* dengan jumlah populasi. Proses mutasi dapat dilihat pada *Source Code 5.3*.

```

1   OffspringM = new double[offspringMutasi][6];
2   int chooseGen[] = new int[2];
3   int mulai = 0;
4   while (mulai < offspringMutasi) {
5       int kromosom_mutation = random_mr[mulai] - 1;

```

```

6     for (int i = 0; i < 6; i++) {
7         OffspringM[mulai][i]=kromosom_awal[kromosom_mutation][i];
8     }
9     ArrayList<Integer> acakgen = new ArrayList<>();
10    for (int i = 1; i <= 6; i++) {
11        acakgen.add(i);
12    }
13    Collections.shuffle(acakgen);
14    System.out.print("Gen terpilih  : ");
15    for (int i = 0; i < 2; i++) {
16        chooseGen[i] = acakgen.get(i);
17        System.out.print(chooseGen[i] + " ");
18    }
19    mulai++;
20 }
21 System.out.println("");
22 System.out.println("Hasil Mutasi");
23 for (int i = 0; i < OffspringM.length; i++) {
24     double gen1 = OffspringM[i][chooseGen[0] - 1];
25     double gen2 = OffspringM[i][chooseGen[1] - 1];
26     OffspringM[i][chooseGen[1] - 1] = gen1;
27     OffspringM[i][chooseGen[0] - 1] = gen2;
28     for (int j = 0; j < 6; j++) {
29         System.out.print(OffspringM[i][j] + " ");
30     }
31     System.out.println("");
32 }

```

Source Code 5.3 Proses Mutasi

Penjelasan *source code* proses mutasi adalah sebagai berikut:

1. Baris 3 membuat variabel *array* 2 dimensi untuk menyimpan jumlah *offspring*.
2. Baris 4-20 merupakan proses pemilihan gen dari kromosom *parent* yang akan dilakukan pertukaran nilai gen.
3. Baris 21-32 merupakan proses penukaran gen untuk dilakukan mutasi.

5.1.4 Implementasi Perhitungan Nilai *Fitness*

Perhitungan *fitness* dilakukan pada setiap individu dengan rumus 10000 dibagi dengan total jumlah harga setiap kromosom ditambah dengan hasil kali penalti dengan konstanta, yaitu 10.000. Proses perhitungan *fitness* dapat dilihat pada *Source Code 5.4*.

```

1     for (int j = 0; j < kromosom.length; j++) {
2         for (int k = 0; k < 6; k++) {
3             hargatot[j]+=kromosom[j][k]*ketersediaan_nutrisi_pakan[k][4];
4         }
5     }
6     for (int i = 0; i < kromosom.length; i++) {
7         total_kromosom = 0;
8         total_pakan_serat = 0;
9         total_pakan_konsentrat = 0;
10        minimum_serat = getBerat() * 0.01;
11        maksimum_konsentrat = getBerat() * 0.005;
12        penalty[i] = 0;
13    }

```

```

14     for (int j = 0; j < ketersediaan_nutrisi_pakan.length; j++)
15     {
16     nutrisi_DE=kromosom[i][j]*(ketersediaan_nutrisi_pakan[j][0]);
17         if (j < 2) {
18             total_pakan_serat += kromosom[i][j];
19         } else {
20             total_pakan_konsentrat += kromosom[i][j];
21         }
22     totalNutrisiDE[i]+=nutrisi_DE;
23     totalNutrisiProtein[i]+=(kromosom[i][j]*
24         ketersediaan_nutrisi_pakan[j][1] / 100) * 1000;
25     totalNutrisiCa[i]+=(kromosom[i][j]*
26         ketersediaan_nutrisi_pakan[j][2] / 100) * 1000;
27     totalNutrisiP[i]+=(kromosom[i][j]*
28         ketersediaan_nutrisi_pakan[j][3] / 100) * 1000;
29     }
30     if (totalNutrisiDE[i] < kebutuhan_nutrisi_pakan[0]) {
31         penalty[i]+=(kebutuhan_nutrisi_pakan[0]-
32             totalNutrisiDE[i]);}
33     if (totalNutrisiProtein[i]< kebutuhan_nutrisi_pakan[1]) {
34         penalty[i]+=(kebutuhan_nutrisi_pakan[1]-
35             totalNutrisiProtein[i]);}
36     if (totalNutrisiCa[i] < kebutuhan_nutrisi_pakan[2]) {
37         penalty[i]+=(kebutuhan_nutrisi_pakan[2]-
38             totalNutrisiCa[i]);}
39     if (totalNutrisiP[i] < kebutuhan_nutrisi_pakan[3]) {
40         penalty[i]+=(kebutuhan_nutrisi_pakan[3]-
41             totalNutrisiP[i]);}
42     if (total_pakan_serat < minimum_serat) {
43         penalty[i]+=(minimum_serat-total_pakan_serat);}
44     if (total_pakan_konsentrat > maksimum_konsentrat) {
45         penalty[i]+=(total_pakan_konsentrat-
46             maksimum_konsentrat);}
47     fitness[i] = hitungFitness(penalty[i], hargatot[i]);
48     }
49     private double hitungFitness(double penalty, int harga) {
50         double fitness = 0;
51         fitness = 10000 / (harga + (penalty * 10000));
52         return fitness;
53     }

```

Source Code 5.4 Perhitungan Nilai *Fitness*

Penjelasan *source code* proses perhitungan nilai *fitness* adalah sebagai berikut:

1. Baris 1-6 merupakan proses perulangan untuk mendapatkan harga setiap individu.
2. Baris 6-12 merupakan proses menghitung kebutuhan serat dan konsentrat.
3. Baris 13-28 merupakan proses menghitung total nutrisi dari setiap bobot bahan pakan.
4. Baris 29-45 merupakan proses untuk menghitung nilai *penalty*.
5. Baris 48-52 merupakan proses perhitungan nilai *fitness* dengan rumus yang telah ditetapkan dengan konstanta tertentu.

5.1.5 Implementasi Proses Seleksi

Proses seleksi yang dilakukan dengan cara mengurutkan nilai *fitness* dari yang terbesar hingga yang terkecil. Kemudian individu yang memiliki nilai *fitness* tertinggi sebanyak jumlah *popSizenya* akan terpilih sebagai individu berikutnya yang akan di proses kedalam reproduksi selanjutnya. Semakin tinggi nilai *fitnessnya* semakin besar pula peluang terpilihnya individu tersebut. Proses seleksi *elitism* yang digunakan dapat dilihat pada *Source Code 5.5*.

```

1 private void seleksi(double[][] kromosombaru) {
2 double kromosom_urut[][]=new
3 double[kromosombaru.length][9];
4 double max[] = new double[9];
5 int k;
6 int l;
7 int m;
8 boolean isi;
9 for (int i = 0; i < kromosombaru.length; i++) {
10 kromosom_urut[i][0] = kromosombaru[i][0];
11 kromosom_urut[i][1] = kromosombaru[i][1];
12 kromosom_urut[i][2] = kromosombaru[i][2];
13 kromosom_urut[i][3] = kromosombaru[i][3];
14 kromosom_urut[i][4] = kromosombaru[i][4];
15 kromosom_urut[i][5] = kromosombaru[i][5];
16 kromosom_urut[i][6] = hargatot[i];
17 kromosom_urut[i][7] = penalty[i];
18 kromosom_urut[i][8] = fitness[i];
19 }
20 for (k = 0; k < kromosom_urut.length; k++) {
21 isi = false;
22 int maxKey = 0;
23 for (m = k; m < kromosom_urut.length; m++) {
24 if (isi == false || kromosom_urut[m][8] >
25 max[8]) {
26 maxKey = m;
27 max[8] = kromosom_urut[m][8];
28 max[0] = kromosom_urut[m][0];
29 max[1] = kromosom_urut[m][1];
30 max[2] = kromosom_urut[m][2];
31 max[3] = kromosom_urut[m][3];
32 max[4] = kromosom_urut[m][4];
33 max[5] = kromosom_urut[m][5];
34 max[6] = kromosom_urut[m][6];
35 max[7] = kromosom_urut[m][7];
36 }
37 isi = true;
38 }
39 kromosom_urut[maxKey][8] = kromosom_urut[k][8];
40 kromosom_urut[maxKey][0] = kromosom_urut[k][0];
41 kromosom_urut[maxKey][1] = kromosom_urut[k][1];
42 kromosom_urut[maxKey][2] = kromosom_urut[k][2];
43 kromosom_urut[maxKey][3] = kromosom_urut[k][3];
44 kromosom_urut[maxKey][4] = kromosom_urut[k][4];
45 kromosom_urut[maxKey][5] = kromosom_urut[k][5];
46 kromosom_urut[maxKey][6] = kromosom_urut[k][6];
47 kromosom_urut[maxKey][7] = kromosom_urut[k][7];
48 kromosom_urut[k][8] = max[8];

```

```

49     kromosom_urut[k][0] = max[0];
50     kromosom_urut[k][1] = max[1];
51     kromosom_urut[k][2] = max[2];
52     kromosom_urut[k][3] = max[3];
53     kromosom_urut[k][4] = max[4];
54     kromosom_urut[k][5] = max[5];
55     kromosom_urut[k][6] = max[6];
56     kromosom_urut[k][7] = max[7];
57 }
58 for (int i = 0; i < kromosom_urut.length; i++) {
59     for (int j = 0; j < 7; j++) {
60         System.out.print(kromosom_urut[i][j] + " ");
61         System.out.println();
62     }
63     for (int i = 0; i < kromosom_awal.length; i++) {
64         for (int j = 0; j < 9; j++) {
65             if (j < 6) {
66                 kromosom_awal[i][j] = kromosom_urut[i][j];
67             } else if (j == 6) {
68             } else if (j == 7) {
69                 penalty[i] = kromosom_urut[i][j];
70             } else if (j == 8) {
71                 fitness[i] = kromosom_urut[i][j];}}g++;}}

```

Source Code 5.5 Proses Seleksi

Penjelasan *source code* proses seleksi adalah sebagai berikut:

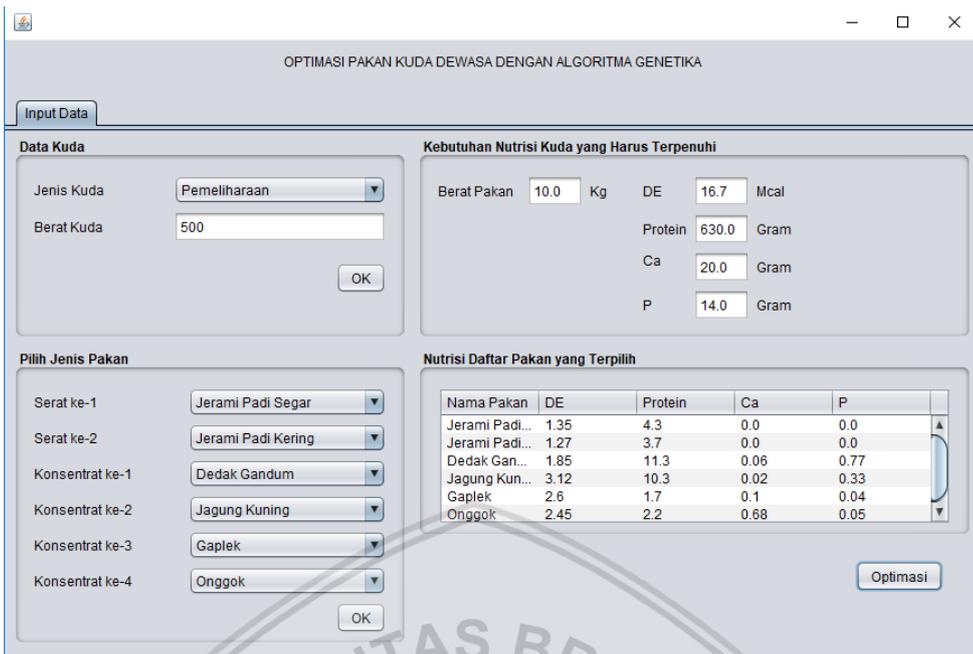
1. Baris 9-19 merupakan proses perulangan untuk memanggil kromosom tiap individu *parent* dan *offspring*.
2. Baris 20-57 merupakan proses mengurutkan seluruh individu berdasarkan nilai fitness yang terbesar.
3. Baris 58-71 merupakan proses memanggil kromosom baru yang telah diurutkan sebanyak populasi yang telah ditentukan.

5.2 Implementasi Antarmuka

Implementasi antarmuka program komposisi pakan kuda menggunakan algoritme genetika yang akan digunakan oleh user untuk berinteraksi. Berikut hasil implementasi antarmuka pada masing-masing halaman pada aplikasi ini.

5.2.1 Implementasi Halaman *Input Data Kuda dan Bahan Pakan*

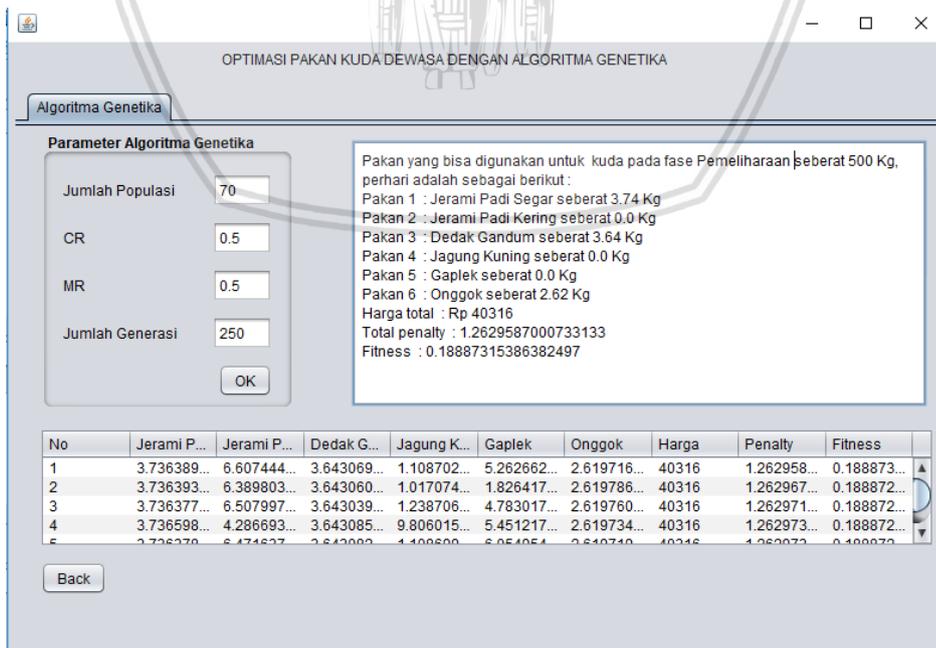
Halaman ini merupakan halaman awal ketika user masuk ke dalam sistem. Halaman *input data kuda dan bahan pakan* terdiri atas 2 bagian yaitu bagian untuk menghitung nutrisi dan untuk menampilkan kandungan bahan pakan. Bagian untuk menghitung nilai nutrisi yang dibutuhkan adalah dengan cara memasukkan jenis kuda dan berat badan. Untuk mendapatkan nilai nutrisi setiap bahan pakan maka terlebih dahulu untuk memilih pakan yang diinginkan. Tampilan halaman ini dapat dilihat pada Gambar 5.1.



Gambar 5.1 Halaman *Input Data* Kuda dan Bahan Pakan

5.2.2 Implementasi Halaman Algoritme Genetika

Pada halaman algoritme genetika ini terjadi proses perhitungan untuk mendapatkan hasil rekomendasi terbaik. Untuk mendapatkan rekomendasi terbaik dengan perhitungan algoritme genetika, terlebih dahulu memasukkan jumlah populasi, nilai *crossover rate*, nilai *mutation rate* dan jumlah generasi untuk perhitungan. Tampilan menu halaman algoritme genetika dapat dilihat pada Gambar 5.2



Gambar 5.2 Halaman Algoritme Genetika



BAB 6 PENGUJIAN

Pada bab pengujian ini akan dilakukan pengujian parameter terhadap algoritme genetika yang di implementasikan. Pengujian dilakukan berdasarkan skenario perancangan pada bab perancangan.

6.1 Sistematika Pengujian

Pengujian dilakukan untuk mengetahui parameter algoritme genetika yang dianggap optimal dalam menghasilkan nilai *fitness* terbaik sehingga dapat ditemukan parameter optimal yang digunakan untuk menyelesaikan permasalahan optimasi komposisi pakan untuk kuda dewasa. Pengujian ini dilakukan dengan masukan sebagai berikut:

- a. Data kuda
 - Jenis kuda : Pemeliharaan
 - Berat badan : 500 kg
- b. Data Bahan Pakan
 - Jenis bahan pakan yang digunakan dalam setiap proses pengujian adalah jerami padi segar, dedak gandum, bungkil kelapa, jagung kuning, rendeng kering, rumput ilalang.

6.2 Pengujian Nilai Populasi

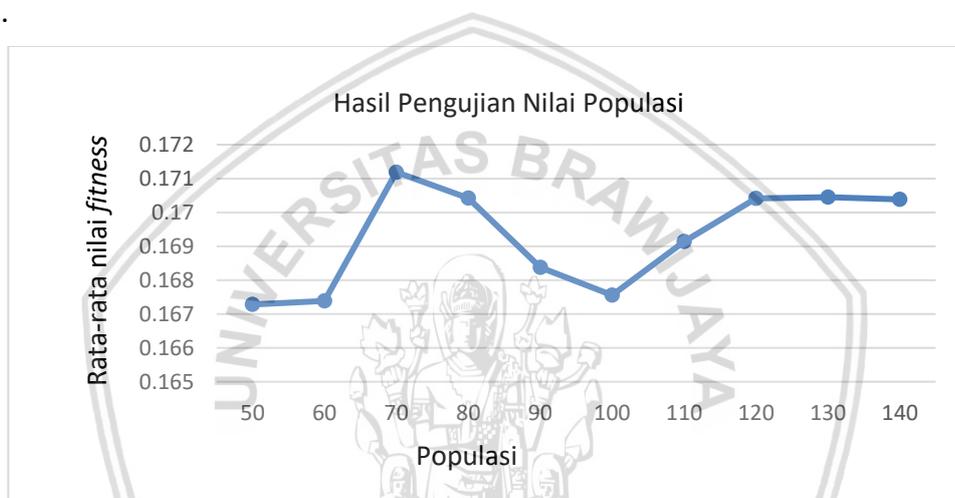
Pengujian nilai populasi adalah pengujian yang bertujuan untuk dapat mengetahui jumlah atau ukuran populasi yang tepat sehingga mampu memberikan solusi yang optimal dalam suatu masalah, termasuk kasus untuk optimasi komposisi pakan kuda dewasa. Ukuran populasi yang digunakan pada kasus ini adalah kelipatan 10 dimulai dari 10 sampai dengan 10 sebanyak 10 kali percobaan. Pada pengujian ini digunakan nilai generasi = 50, $cr = 0,2$ dan $mr = 0,2$. Pengujian nilai populasi ditunjukkan pada Tabel 6.1.

Tabel 6.1 Pengujian Nilai Populasi

Ukuran Populasi	Nilai <i>fitness</i>						Rata-rata <i>fitness</i>
	Percobaan ke-						
	1	2	3	4	...	10	
50	0.165793	0.160646	0.165147	0.166725	...	0.168972939	0.167280036
60	0.171169	0.163527	0.172272	0.172901	...	0.160680259	0.167388731
70	0.166627	0.173595	0.170088	0.163247	...	0.175661017	0.17119007
80	0.17051	0.175179	0.171769	0.165858	...	0.171082593	0.17042712
90	0.169459	0.163992	0.174279	0.169042	...	0.16474631	0.168375225
100	0.169017	0.161164	0.168292	0.170669	...	0.16430943	0.167552097
110	0.161064	0.162829	0.169723	0.169399	...	0.176174375	0.169143469

Ukuran Populasi	Nilai <i>fitness</i>						Rata-rata <i>fitness</i>
	Percobaan ke-						
	1	2	3	4	...	10	
120	0.170121	0.163873	0.171926	0.182077	...	0.172242701	0.170408531
130	0.168247	0.167955	0.170194	0.171325	...	0.173645782	0.170448472
140	0.165727	0.172084	0.171694	0.169642	...	0.169370131	0.170389269

Dari hasil uji coba pada Tabel 6.1 diketahui bahwa semakin besar ukuran populasi maka akan menghasilkan nilai *fitness* yang semakin baik hingga mencapai titik populasi yang terbaik pada ukuran 70 seperti yang ditunjukkan pada Gambar 6.1.



Gambar 6.1 Pengujian Nilai Populasi

Berdasarkan Gambar 6.1 dapat disimpulkan bahwa ukuran populasi yang kecil dapat menghalangi keragaman individu yang dibentuk dan sempitnya area eksplorasi untuk menghasilkan *offspring* dari proses reproduksi. Nilai *fitness* terbaik dari 10 kali percobaan dari ukuran populasi 50 hingga 140 adalah ukuran populasi 70 dengan rata-rata nilai *fitness* 0.17119007 dan mengalami penurunan pada ukuran populasi 80. Tidak ada kenaikan signifikan yang terjadi setelah ukuran populasi 110 dan bisa dikatakan konvergen sehingga ukuran populasi yang lebih besar tidak menjamin akan memberikan hasil yang lebih baik.

6.3 Pengujian Nilai Generasi

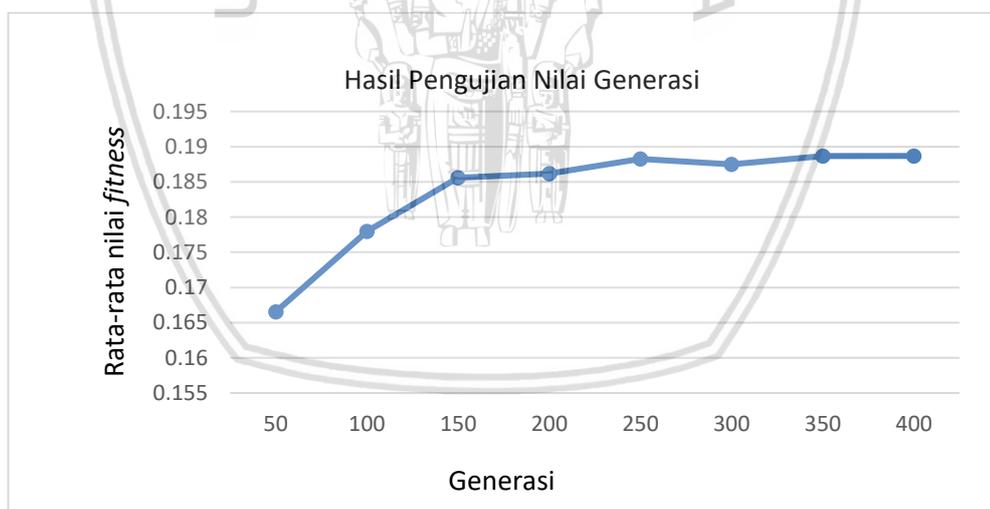
Pengujian nilai generasi adalah pengujian yang bertujuan untuk mengetahui jumlah atau ukuran generasi yang tepat sehingga mampu memberikan solusi yang optimal dalam menyelesaikan suatu masalah. Ukuran generasi yang digunakan serta untuk diujikan pada kasus ini adalah kelipatan 50 dimulai dari 50 sampai 400 sebanyak 10 kali percobaan. Pada pengujian ini digunakan nilai populasi terbaik yang didapatkan pada pengujian sebelumnya yaitu = 70, $cr = 0,2$ dan $mr = 0,2$. Pengujian nilai generasi ditunjukkan pada Tabel 6.2.



Tabel 6.2 Pengujian Nilai Generasi

Ukuran Generasi	Nilai <i>fitness</i>						Rata-rata <i>fitness</i>
	Percobaan ke-						
	1	2	3	4	...	10	
50	0.167413	0.162438	0.150749	0.168786	...	0.164408	0.166543
100	0.176756	0.172162	0.180273	0.177725	...	0.183220	0.177969
150	0.185814	0.182184	0.186965	0.185870	...	0.185445	0.185559
200	0.188222	0.185705	0.187519	0.188229	...	0.185012	0.186154
250	0.186826	0.188300	0.188691	0.188572	...	0.188672	0.188259
300	0.188605	0.188038	0.187706	0.185581	...	0.188393	0.187507
350	0.188774	0.188816	0.188703	0.188007	...	0.188419	0.188659
400	0.187954	0.188819	0.188723	0.188737	...	0.188845	0.188678

Dari hasil uji coba pada Tabel 6.2 diketahui bahwa semakin besar ukuran generasi maka akan menghasilkan nilai *fitness* yang semakin baik hingga mencapai titik populasi yang stabil pada ukuran generasi 250 seperti yang ditunjukkan pada Gambar 6.1.



Gambar 6.2 Pengujian Nilai Generasi

Berdasarkan Gambar 6.2 dapat diketahui bahwa nilai *fitness* terendah didapatkan pada ukuran generasi 50 karena jumlah ukuran generasi tersebut masih tidak mencukupi untuk mendapatkan hasil yang optimal. Sehingga disimpulkan bahwa nilai *fitness* terbaik dari 10 kali percobaan dari ukuran generasi 50 hingga 400 adalah ukuran generasi 250 dengan rata-rata nilai *fitness* 0.188259872, karena sulit diduplikasinya nilai *fitness* yang lebih baik setelah ukuran generasi 300. Tidak ada kenaikan signifikan yang terjadi setelah ukuran

generasi tersebut dan bisa dikatakan konvergen sehingga ukuran generasi yang lebih besar tidak menjamin akan memberikan hasil yang lebih baik.

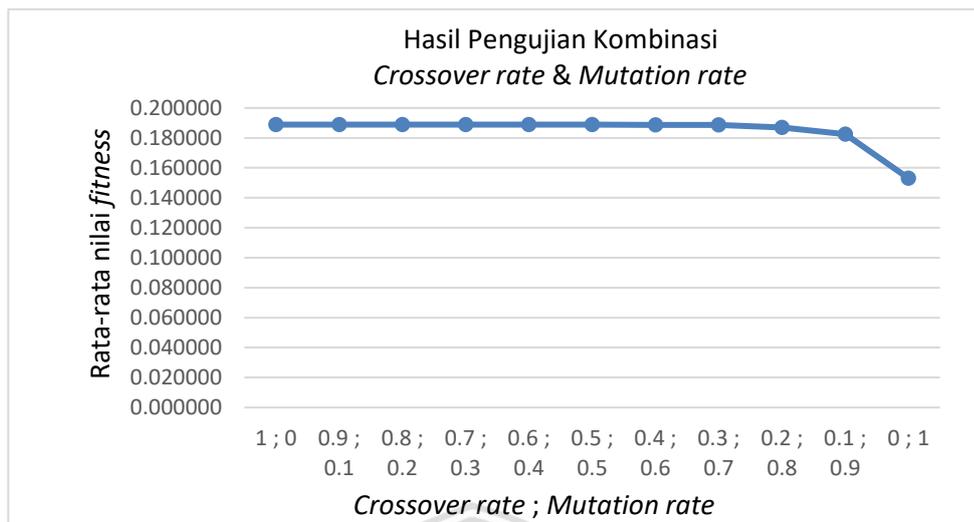
6.4 Pengujian Kombinasi *Crossover rate* & *Mutation rate*

Pengujian kombinasi *crossover rate* dan *mutation rate* adalah pengujian yang bertujuan untuk mengetahui jumlah atau ukuran kombinasi *crossover rate* dan *mutation rate* yang tepat sehingga mampu memberikan solusi yang optimal dalam menyelesaikan suatu masalah. Terdapat 10 kombinasi *crossover rate* dan *mutation rate* yang digunakan serta diujikan pada kasus ini adalah sebanyak 10 kali. Pada pengujian ini digunakan ukuran populasi optimal pada pengujian sebelumnya yaitu ukuran populasi sebanyak 70 dan ukuran generasi sebanyak 250. Pengujian kombinasi *crossover rate* dan *mutation rate* dapat dilihat pada Tabel 6.3.

Tabel 6.3 Pengujian Kombinasi *Crossover rate* & *Mutation rate*

Nilai Cr	Nilai Mr	Nilai <i>fitness</i>						Rata-rata <i>fitness</i>
		Percobaan ke-						
		1	2	3	4	...	10	
1	0	0.188875	0.188878	0.188855	0.188878	...	0.188876	0.188792
0.9	0.1	0.188877	0.188878	0.188876	0.188872	...	0.188871	0.188875
0.8	0.2	0.188874	0.188876	0.188877	0.188878	...	0.188876	0.188876
0.7	0.3	0.188878	0.188878	0.18887	0.188874	...	0.188874	0.188874
0.6	0.4	0.188877	0.188877	0.188877	0.188876	...	0.188875	0.188876
0.5	0.5	0.25002	0.188878	0.188873	0.188875	...	0.188897	0.188881
0.4	0.6	0.188875	0.188477	0.188863	0.188738	...	0.18851	0.188693
0.3	0.7	0.188553	0.188693	0.188815	0.188473	...	0.188846	0.188676
0.2	0.8	0.188757	0.178973	0.188712	0.18864	...	0.188787	0.18691
0.1	0.9	0.18822	0.179443	0.181049	0.18594	...	0.169272	0.182507
0	1	0.164387	0.148725	0.152239	0.149458	...	0.149627	0.153091

Dari hasil uji coba pada Tabel 6.3 diketahui bahwa kombainai *cr* dan *mr* terbaik adalah 0.5 dan 0.5 dengan rata-rata nilai *fitness* 0.188881 seperti yang ditunjukkan pada Gambar 6.1.



Gambar 6.3 Pengujian Kombinasi *Crossover rate* dan *Mutation rate*

Berdasarkan Gambar 6.3 dapat diketahui bahwa tidak ada perubahan yang signifikan terjadi antara nilai *cr* dan *mr* 1 & 0 dan 0,3 & 0,7. Sedangkan mengalami penurunan semakin rendahnya nilai *cr*. Hasil pengujian *crossover rate* dan *mutation rate* menunjukkan grafik dengan perubahan yang tidak begitu signifikan atau bisa disebut konvergen menandakan bahwa hasil yang diperoleh dari parameter ukuran populasi dan ukuran generasi terbaik yang diperoleh dari pengujian sebelumnya telah mampu memberikan hasil yang optimal meskipun diberikan kombinasi *crossover rate* dan *mutation rate* yang lain. *Crossover rate* yang terlalu besar dan *mutation rate* yang terlalu kecil akan menurunkan kemampuan algoritme genetika untuk menginvestigasi area baru pada ruang pencarian yang berguna untuk meningkatkan keragaman populasi. Dan *crossover rate* yang kecil dan *mutation rate* yang besar akan menurunkan kemampuan algoritme genetika untuk melakukan eksploitasi area *optimum local* (Mahmudy, 2014). Pada pengujian yang dihasilkan diketahui bahwa dengan nilai *cr* yang semakin kecil dan nilai *mr* yang semakin besar membuat nilai rata-rata *fitness* yang diperoleh semakin menurun. Sehingga eksploitasi memiliki pengaruh yang signifikan dibandingkan dengan eksplorasinya.

6.5 Analisis Hasil Pengujian

Untuk mengetahui ketercukupan nutrisi dari kromosom dan harga minimum yang didapatkan pada proses algoritme genetika maka dilakukan analisis. Pada penelitian ini didapatkan parameter terbaik dengan nilai ukuran populasi = 70, jumlah generasi = 250, dan kombinasi *cr* dan *mr* yaitu 0,5 dan 0,5. Hasil pengujian dengan parameter terbaik dapat dilihat pada Tabel 6.4, Tabel 6.5, Tabel 6.6, Tabel 6.7.

Dengan menggunakan parameter tersebut didapatkan nilai *fitness* terbaik yaitu 0.1888 dengan nilai pelanggaran 1.26. Berdasarkan wawancara dengan pakar, dari hasil rekomendasi yang diberikan untuk pemberian pakan kuda dewasa pada fase pemeliharaan seberat 500 kg dengan harga Rp. 40365 yang terdiri dari



3.736 kg jerami padi segar, 3.646 kg dedak gandum, dan 2.6217 kg onggok. Biaya yang dihasilkan lebih murah dibandingkan dengan biaya yang biasa digunakan peternak sekitar Rp. 70000. Total pakan yang dihasilkan sistem berjumlah 10 kilogram dan tidak melewati batas maksimum pemberian pakan harian dengan nutrisi yang tersedia sebesar 18.2 Mcal *Digestible Energy*, 629.8 gr Protein, 20 gr Kalsium, dan 29 gr Fosfor, mampu memenuhi kebutuhan nutrisi kuda tersebut. Adanya nilai pelanggaran yang muncul menandakan bobot serat yang diberikan kurang dari syarat minimum pemberian serat, tetapi kondisi berikut masih bisa menjadi jumlah pakan yang ideal bila kuda tidak memiliki beban kerja yang berat.

Parameter terbaik yang didapatkan selanjutnya digunakan untuk melakukan pengujian validasi untuk membandingkan hasil rekomendasi pakan sistem dengan komposisi pemberian pakan yang selama diberikan oleh peternak pada kuda dewasa fase pemeliharaan dengan berat 500 kilogram. Komposisi bahan pakan pertama terdiri dari jerami padi segar, jerami padi kering, dedak gandum, jagung kuning, galek, dan onggok. Hasil pengujian dapat dilihat pada Tabel 6.4.

Tabel 6.4 Pengujian Komposisi 1 dengan Parameter Terbaik

Jerami Padi Segar	Jerami Padi Kering	Dedak Gandum	Jagung Kuning	Galek	Onggok	Harga	Penalty	Fitness	Harga Awal
3.736	0.0	3.646	0.0	0.0	2.617	40365	1.26	0.1888	70000
3.736	0.0	3.646	0.0	0.0	2.617	40365	1.26	0.1888	
3.736	0.0	3.646	0.0	0.0	2.617	40365	1.26	0.1888	

Pada Tabel 6.4 menunjukkan bahwa hasil rekomendasi pakan oleh sistem memberikan harga Rp. 40365. Harga yang diberikan lebih murah daripada harga yang selama ini dibutuhkan oleh peternak yaitu sekitar Rp. 70000.

Komposisi bahan pakan kedua terdiri dari rumput gajah, lamtoro segar, galek, bungkil kedelai, tetes, dan bungkil kelapa sawit. Hasil pengujian dapat dilihat pada Tabel 6.5.

Tabel 6.5 Pengujian Komposisi 2 dengan Parameter Terbaik

Rumput Gajah	Lamtoro Segar	Galek	Bungkil Kedelai	Tetes	Bungkil Kelapa Sawit	Harga	Penalty	fitness	Harga Awal
0.0	5.0	0.68	0.0	0.07	4.25	40375	0.0	0.24768	60000
0.0	5.0	0.68	0.0	0.07	4.25	40375	0.0	0.24768	
0.0	5.0	0.68	0.0	0.07	4.25	40375	0.0	0.24768	

Pada Tabel 6.5 menunjukkan bahwa hasil rekomendasi pakan oleh sistem memberikan harga Rp. 40375. Harga yang diberikan lebih murah daripada harga yang selama ini dibutuhkan oleh peternak yaitu sekitar Rp. 60000.

Komposisi bahan pakan ketiga terdiri dari jerami jagung, rumput ilalang, bungkil kacang, tetes, jagung kuning, dan galek. Hasil pengujian dapat dilihat pada Tabel 6.6.

Tabel 6.6 Pengujian Komposisi 3 dengan Parameter Terbaik

Jerami Jagung	Rumput Ilalang	Bungkil Kacang	Tetes	Jagung Kuning	Gaplek	Harga	Penalty	fitness	Harga Awal
5.0	0.0	0.81	2.86	0.0	1.33	30810	0.0	0.32457	45000
5.0	0.0	0.81	2.86	0.0	1.33	30810	0.0	0.32457	
5.0	0.0	0.81	2.86	0.0	1.33	30810	0.0	0.32457	

Pada Tabel 6.6 menunjukkan bahwa hasil rekomendasi pakan oleh sistem memberikan harga Rp. 30810. Harga yang diberikan lebih murah daripada harga yang selama ini dibutuhkan oleh peternak yaitu sekitar Rp. 45000.

Komposisi bahan pakan keempat terdiri dari daun ketela pohon segar, lamtoro segar, dedak gandum, jagung kuning, bungkil kedelai, dan bungkil kelapa. Hasil pengujian dapat dilihat pada Tabel 6.7.

Tabel 6.7 Pengujian Komposisi 4 dengan Parameter Terbaik

Daun Ketela Pohon Segar	Lamtoro Segar	Dedak Gandum	Jagung Kuning	Bungkil Kedelai	Bungkil Kelapa	Harga	Penalty	fitness	Harga Awal
5.0	0.0	0.0	0.0	0.0	5.0	39980	0.0	0.25013	57000
5.0	0.0	0.0	0.0	0.0	5.0	39980	0.0	0.25013	
5.0	0.0	0.0	0.0	0.0	5.0	39980	0.0	0.25013	

Pada Tabel 6.7 menunjukkan bahwa hasil rekomendasi pakan oleh sistem memberikan harga Rp. 39980. Harga yang diberikan lebih murah daripada harga yang selama ini dibutuhkan oleh peternak yaitu sekitar Rp. 57000.

BAB 7 PENUTUP

Pada bab penutup ini membahas tentang hasil yang diperoleh dari implementasi dan pengujian algoritme genetika pada optimasi komposisi pakan kuda dewasa.

7.1 Kesimpulan

Berdasarkan hasil penelitian mengenai Optimasi Pakan Kuda Dewasa Menggunakan Algoritme Genetika dapat disimpulkan bahwa:

1. Implementasi algoritme genetika pada permasalahan optimasi pakan untuk memenuhi standar nutrisi kuda dewasa dengan kromosom sepanjang 6 gen yang merupakan representasi dari 2 serat dan 4 konsentrat dengan menggunakan representasi kromosom secara *real code*, metode *extended intermediate*, *reciprocal exchange mutation*, serta seleksi dengan metode *elitism selection*.
2. Menentukan parameter algoritme genetika didapatkan dengan melakukan pengujian ukuran populasi, ukuran generasi, serta kombinasi *crossover rate* dan *mutation rate*. Ketiga parameter yang diujikan memiliki pengaruh terhadap nilai *fitness* yang didapatkan. Pengujian ukuran populasi yang optimal berada pada ukuran populasi 70 dengan nilai *fitness* 0.17119007. Sehingga dapat disimpulkan bahwa ukuran populasi mempengaruhi nilai *fitness* yang diperoleh, namun semakin besar ukuran populasi tidak menjamin didapkannya solusi yang lebih baik. Pengujian jumlah generasi yang optimal berada pada generasi 250 dengan nilai *fitness* 0.188259872. Untuk jumlah generasi yang lebih besar menunjukkan perubahan hasil yang tidak begitu signifikan namun semakin besar generasi yang digunakan maka akan memerlukan waktu yang lebih lama untuk proses komputasi. Pengujian kombinasi *crossover rate* dan *mutation rate* terbaik adalah 0.5 & 0.5 dengan nilai *fitness* 0.188881.
3. Tingkat kualitas solusi didapatkan berdasarkan perhitungan nilai *fitness* yang dihasilkan. *Fitness* tertinggi yang didapatkan berdasarkan pengujian yang dilakukan adalah individu yang memiliki solusi yang paling optimal memiliki nilai *penalty* yang kecil ataupun tidak memiliki nilai sama sekali. Pada pengujian didapatkan nilai *fitness* tertinggi sebesar 0.188259872 dan memiliki *penalty* 1.2636659122167355 dengan harga Rp 40316.

7.2 Saran

Beberapa saran yang bisa diberikan sebagai bahan pertimbangan penelitian selanjutnya yang lebih baik berdasarkan penelitian Optimasi Komposisi Pakan Kuda Dewasa Menggunakan Algoritme Genetika yaitu:

1. Dapat ditambahkan kebutuhan nutrisi lain yang lebih kompleks selain energi metabolisme, kalsium, protein, dan fosfor.

2. Dapat ditambahkan fase-fase kuda yang lebih luas lagi selain fase pemeliharaan, 90 hari sebelum melahirkan, menyusui 3 bulan awal, dan menyusui 3 bulan akhir.
3. Dapat ditambahkan jenis beban kerja kuda yang akan mempengaruhi ratio kebutuhan serat dan konsentrat.
4. Pada penelitian ini menggunakan kombinasi 6 pakan yang terdiri dari 2 jenis serat dan 4 jenis konsentrat yang berbeda. Kombinasi dapat ditingkatkan dengan harapan mendapatkan harga yang lebih optimal.
5. Pada penelitian ini menggunakan metode mutasi *reciprocal exchange* yang sederhana, kedepannya dapat menggunakan metode mutasi selain *reciprocal exchange* agar mendapatkan populasi kromosom yang lebih beragam.



DAFTAR PUSTAKA

- Akoso, B., T. 1996. *Kebugaran atau kesehatan Sapi*. Kanisius. Yogyakarta
- Berndtsson, M., Hansson, J., Olsson, B. & Lundell, B., 2008. *Thesis projects: a guide for students in Computer Science and Information Systems*. 2nd ed. London: Springer-Verlag London Limited.
- Fakhiroh, D, Mahmudy, WF dan Indriati, 2015. *Optimasi komposisi pakan sapi perah menggunakan algoritma genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 14.
- Hammer, D. 1993. *Care of the Stable Horse*. B. T. Batsford Ltd. London.
- Holcomb, G., H. Kiesling, and G. Lofgreen, 1984. *Digestibility of Diets and Performance by Steers Feed Varying Energy and Protein Level in Feedlot Receiving Program. Livestock Research Beefs and Cattle Growers Shorts Course*. New Mexico State University, Mexico
- Indonesia, K. P. d. k. R., 2013. *Dasar-dasar Pakan Ternak*. Kurikulum 2013 penyunt. Indonesia: Direktorat Pembinaan Sekolah Menengah Kejuruan.
- Mahmudy, W.F., 2013. *Modul Matakuliah Algoritme Evolusi*. PTIIK. Universitas Brawijaya.
- Mahmudy, WF, Marian, RM & Luong, 2014. *Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing system*. ECTI Transactions on Computer and Information Technology (ECTI- CIT), vol. 8, no. 1, pp. 80-93.
- Mansyur, U., H. Tanuwiria, dan D. Rusmana. 2006. *Eksplorasi hijauan pakan kuda dan kandungan nutrisinya*. Seminar Nasional Teknologi Peternakan dan Veteriner. 924-931.
- Margingintyas, E, Mahmudy, WF dan Indriati, 2015. *Penentuan Komposisi Pakan Ternak Untuk Memenuhi Kebutuhan Nutrisi Ayam Petelur Dengan Biaya Minimum Menggunakan Algoritma Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 12.
- Maswarni & Rachman Nofiar. 2014. *Manajemen Pemeliharaan & Pembiakan Kuda*. Penebar Swadaya. Jakarta.
- National Research Council. 2007. *Nutrient Requirements of Horses: Sixth Revised Edition*. Washington, D.C.
- Nugraha, R. A. 2011. *Optimalisasi Formulasi Pakan Ternak Terhadap Ayam Pedaging Dengan Menggunakan Metode Linear Programming*. Universitas Gunadarma.
- Pramesti, D, Mahmudy, WF dan Indriati, 2015. *Optimasi komposisi pakan kambing potong menggunakan algoritme genetik*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 13.

- Pratiwi, MI, Mahmudy, WF & Dewi, C. 2014. *Implementasi Algoritme Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi*, DORO: Respository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol 4, no 6.
- Rahmawati, Putri. 2011. *Pengaruh Pakan Terhadap Performa Reproduksi Kuda Betina di Kuda "Tombo Ati Stable" Salatiga*. Universitas Gadjah Mada.
- Rennard, JP. 2000. *Genetic Algorithms Viewer 1.0*. Tersedia di : <https://www.rennard.org/alife/english/gavgb.html/> [Diakses 8 Maret 2017]
- Saraswati, RP, Mahmudy, WF & Dewi, C. 2016. *Optimasi Komposisi Pakan Ikan Patin Jambal Menggunakan Algoritme Genetika*. PTIIK. Universitas Brawijaya.
- Soeharjono. O. 1990. *Kuda*. Yayasan Pamulang Equistian Center. Jakarta.
- Sudarmono, AS. 2003. *Pedoman Pemeliharaan Ayam Ras Petelur*. Kanisius : Yogyakarta.
- Suyanto, 2014. *Artificial Intelligence (Searching – Reasoning – Planning – Learning) Revisi Kedua*. Bandung: Informatika.
- Tulung, Yohannis Lodewyk R. 2012. *Kebutuhan Energi dan Nutrien Kuda Pacu Indonesia dan Aplikasi pada Formulasi Ransum Berbasis Pakan Lokal*. Institut Pertanian Bogor.
- Wahid, N dan Mahmudy, W.F., 2015. *Optimasi komposisi makanan untuk penderita kolesterol menggunakan algoritme genetika*. DORO: Respository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 15.
- Wahiduddin. 2008. *Ilmu Pakan Ternak ; Bagian 2*. Tersedia di : <https://wah1d.wordpress.com/2008/09/25/ilmu-pakan-ternak-bag2/> [Diakses 8 Maret 2017]
- Widodo, A.W., dan Mahmudy, W.F., 2010. *Penerapan Algoritme Genetika Pada Sistem Rekomendasi Wisata Kuliner*. Jurnal Ilmiah Vol. 5, No. 4, Juli 2010