

repository.ub.ac.id

**OPTIMASI KOMPOSISI PAKAN SAPI PERAH MENGGUNAKAN
IMPROVED PARTICLE SWARM OPTIMIZATION (IPSO)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Fitri Bibi Suryani
NIM: 125150200111076



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016



PENGESAHAN

OPTIMASI KOMPOSISI PAKAN SAPI PERAH MENGGUNAKAN *IMPROVED PARTICLE SWARM OPTIMIZATION* (IPSO)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Fitri Bibi Suryani

NIM: 125150200111076

Skripsi ini telah diuji dan dinyatakan lulus pada
4 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agus Wahyu Widodo, S.T., M.Cs.

NIP: 19740805 200112 1 001

Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D.

NIP: 19720919 199702 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, Agustus 2016

Fitri Bibi Suryani

NIM: 125150200111076



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul "Optimasi Komposisi Pakan Sapi Perah Menggunakan *Improved Particle Swarm Optimization* (IPSO)". Skripsi ini disusun sebagai persyaratan untuk memperoleh gelar Sarjana Komputer di Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam pembuatan skripsi ini tentunya penulis mendapatkan banyak bantuan berupa bimbingan, kritik, saran, doa, dan motivasi dari berbagai pihak. Oleh karena itu pada kesempatan ini, penulis menyampaikan ucapan terima kasih kepada:

1. Agus Wahyu Widodo, S.T., M.Cs., selaku dosen pembimbing pertama yang telah membimbing dan mengarahkan penulis sehingga penulis dapat menyelesaikan skripsi ini.
2. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku dosen pembimbing kedua yang telah membimbing dan mengarahkan penulis sehingga penulis dapat menyelesaikan skripsi ini.
3. Dr. Ir. Irfan H. Djunaidi, M.Sc., selaku dosen Fakultas Peternakan Universitas Brawijaya yang telah meluangkan waktunya untuk berdiskusi dengan penulis terkait komposisi pakan sapi perah.
4. KUD Batu dan para peternak sapi perah di desa Dresel, kecamatan Batu, kota Batu yang telah bersedia memberikan data untuk penelitian ini.
5. Keluarga tercinta yang senantiasa mendukung, memotivasi, dan mendoakan penulis agar skripsi ini dapat terselesaikan dengan baik.
6. Teman – teman seperjuangan Az Zahra Rahma P.A., Arinie Khaqqo, Dewi Maida Safitri, dan Fina Af Idatul Husna yang senantiasa saling memberi dorongan dan motivasi demi terselesaikannya skripsi ini.
7. Dan semua pihak yang tidak bisa disebutkan satu per satu.

Akhir kata "tiada gading yang tak retak", maka atas segala kekurangan dalam skripsi ini, penulis mohon maaf dan mohon dimaklumi. Semoga skripsi ini dapat bermanfaat dan dapat digunakan dengan sebaik-baiknya.

Malang, Agustus 2016

Penulis

fitri.b.suryani@gmail.com

ABSTRAK

Sistem pemberian pakan mempunyai peranan penting dalam usaha sapi perah. Biaya pakan memegang prosentase terbesar yang mencapai 62,5% dari total biaya usaha. Pemberian pakan yang tidak tepat bisa mengakibatkan peternak mengeluarkan biaya yang besar hanya untuk pakan tetapi produksi susu yang dihasilkan tidak maksimal. Untuk itu diperlukan sebuah metode yang dapat menyusun bahan pakan sesuai kebutuhan nutrisi sapi dengan harga pakan yang seminimal mungkin. Metode tersebut adalah algoritma *improved particle swarm optimization* (IPSO). Algoritma IPSO dipilih karena terbukti lebih unggul dari pada algoritma PSO konvensional dan memerlukan tingkat komputasi yang lebih kecil dibandingkan algoritma genetika. Selain mudah untuk diimplementasikan, algoritma IPSO juga dapat menghasilkan solusi optimum lebih efisien dari pada algoritma genetika dan algoritma semut.

Penyelesaian masalah menggunakan algoritma IPSO terdiri dari 6 proses utama yaitu (1) membangun populasi awal; (2) menghitung nilai *fitness* pada tiap partikel; (3) menentukan posisi lokal terbaik (*pbest*); (4) menentukan posisi global terbaik (*gbest*); (5) memperbarui kecepatan partikel; dan terakhir (6) memperbarui posisi partikel. Serangkaian uji coba juga dilakukan untuk mengetahui unjuk kerja dari algoritma IPSO. Dari pengujian parameter IPSO, didapatkan jumlah iterasi optimum sebesar 80, ukuran populasi optimum sebanyak 140, dan interval bobot bahan optimum yaitu 1 sampai 15. Kemudian dari pengujian data, diketahui bahwa algoritma IPSO dapat merekomendasikan komposisi pakan yang lebih murah Rp 16654 dibandingkan cara konvensional. Dan berdasarkan pengujian perbandingan algoritma terbukti bahwa algoritma IPSO dapat menghasilkan solusi optimal dengan harga pakan lebih murah Rp 8317 dari pada menggunakan algoritma PSO. Algoritma IPSO juga terbukti memerlukan jumlah iterasi dan ukuran populasi yang lebih kecil dibandingkan dengan algoritma PSO.

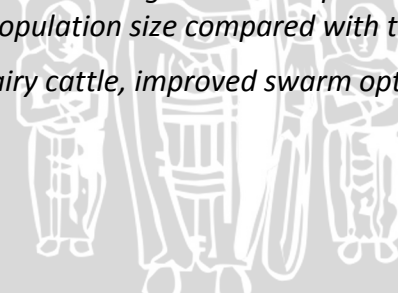
Kata kunci: optimasi, pakan sapi perah, algoritma *improved particle swarm optimization*.

ABSTRACT

Feeding system has an important role in the dairy cattle business. The cost of feed holds the largest percentage which reached 62.5% of total operating expenses. Improper feeding can result in farmers incur huge costs for feed only but the milk production is not optimal. For that we need a method that can compose of feed materials in accordance with the nutritional needs of cattle with minimum feed prices. The method is improved particle swarm optimization algorithm (IPSO). IPSO algorithm chosen because it proved to be superior to conventional PSO algorithm and requires smaller computation than the genetic algorithm. Besides easy to implement, IPSO algorithm also can produce the optimum solution more efficient than the genetic algorithm and ant algorithm.

Problem solving using IPSO algorithm consists of six main processes: (1) establish an initial population; (2) calculate the fitness value of each particle; (3) determine the best local positions (*pbest*); (4) determining the best global position (*gbest*); (5) updating the particle velocity; and finally (6) updating the position of the particle. A series of trials were also conducted to determine the performance of the IPSO algorithm. Based on IPSO parameters testing, obtained the optimum iteration number is 80, the optimum population size is 140, and the weight of materials optimum interval is 1 to 15. Then from the data testing, it is known that the IPSO algorithm can recommend a cheaper feed composition until Rp 16654 compared to the conventional way. And based on comparative testing proved that the algorithm IPSO algorithm can generate the optimal solution with less feed price Rp 8317 than PSO algorithm. IPSO algorithm also proven to require the smaller number of iterations and population size compared with the PSO algorithm.

Keywords: optimization, dairy cattle, improved swarm optimization.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
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 Pakan Sapi Perah.....	8
2.3 Algoritma <i>Improved Particle Swarm Optimization</i> (IPSO).....	11
2.3.1 Algoritma <i>Particle Swarm Optimization</i> (PSO).....	11
2.3.2 Algoritma <i>Improved Particle Swarm Optimization</i> (IPSO)	13
BAB 3 METODOLOGI	16
3.1 Studi Literatur	17
3.2 Analisa Kebutuhan	17
3.3 Pengumpulan Data	17
3.4 Perancangan Sistem.....	17
3.5 Implementasi	18
3.6 Pengujian	19

3.7 Pengambilan Kesimpulan.....	19
BAB 4 PERANCANGAN.....	20
4.1 Formulasi Permasalahan.....	20
4.1.1 Menghitung Kebutuhan Nutrisi Sapi	20
4.2 Alur Penyelesaian Masalah Menggunakan Algoritma IPSO.....	21
4.2.1 Inialisasi Variabel.....	22
4.2.2 Membangun Populasi Awal	22
4.2.3 Menghitung <i>Fitness</i>	23
4.2.4 Menentukan Posisi Lokal Terbaik	24
4.2.5 Menentukan Posisi Global Terbaik	25
4.2.6 Memperbarui Kecepatan Partikel.....	26
4.2.7 Memperbarui Posisi Partikel.....	26
4.3 Contoh Penyelesaian Masalah Menggunakan Algoritma IPSO	27
4.3.1 Inialisasi Variabel	27
4.3.2 Menghitung Kebutuhan Nutrisi Sapi Perah	27
4.3.3 Membangun Populasi Awal	28
4.3.4 Menghitung <i>Fitness</i>	29
4.3.5 Menentukan Posisi Lokal Terbaik	33
4.3.6 Menentukan Posisi Global Terbaik	34
4.3.7 Memperbarui Kecepatan Partikel.....	34
4.3.8 Memperbarui Posisi Partikel.....	35
4.4 Perancangan Pengujian	35
4.4.1 Perancangan Pengujian Jumlah Iterasi	36
4.4.2 Perancangan Pengujian Ukuran Populasi	36
4.4.3 Perancangan Pengujian Interval Berat Bahan Pakan.....	37
4.5 Perancangan Basis Data.....	37
4.6 Perancangan Antarmuka	38
4.6.1 Perancangan Antarmuka <i>Input</i> Data.....	38
4.6.2 Perancangan Antarmuka Pengelolaan Data Bahan Pakan	39
4.6.3 Perancangan Antarmuka <i>Input</i> dan <i>Edit</i> Data Bahan Pakan.....	40
BAB 5 IMPLEMENTASI	42
5.1 Implementasi Algoritma	42

5.1.1 Implementasi Membangun Populasi Awal	42
5.1.2 Implementasi Menghitung <i>Fitness</i>	43
5.1.3 Implementasi Menentukan Posisi Lokal Terbaik	44
5.1.4 Implementasi Menentukan Posisi Global Terbaik	44
5.1.5 Implementasi Memperbarui Kecepatan Partikel.....	45
5.1.6 Implementasi Memperbarui Posisi Partikel.....	46
5.2 Implementasi Basis Data.....	46
5.3 Implementasi Antarmuka	46
5.3.1 Implementasi Antarmuka <i>Input</i> Data	47
5.3.2 Implementasi Antarmuka Pengelolaan Data Bahan Pakan	47
5.3.3 Implementasi Antarmuka <i>Input</i> dan <i>Edit</i> Data Bahan Pakan	48
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	50
6.1 Pengujian Parameter IPSO.....	50
6.1.1. Pengujian Jumlah Iterasi	50
6.1.2. Pengujian Ukuran Populasi	51
6.1.3. Pengujian Interval Berat Bahan Pakan.....	53
6.2 Pengujian Data.....	54
6.3 Pengujian Perbandingan Algoritma	56
6.4 Pembahasan Hasil Pengujian.....	58
BAB 7 Penutup	59
7.1 Kesimpulan.....	59
7.2 Saran	59
DAFTAR PUSTAKA.....	60
LAMPIRAN A Hasil Wawancara Narasumber	62
LAMPIRAN B Hasil Observasi Data Penerimaan Susu Di KUD Batu dan Data Pemberian Pakan Peternak Di Desa Dresel, Kecamatan Batu, Kota Batu	63

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	7
Tabel 2.2 Kebutuhan nutrisi untuk hidup pokok sapi perah laktasi	9
Tabel 2.3 Kebutuhan nutrisi untuk produksi 1 liter susu	9
Tabel 2.4 Kebutuhan NE _L per kg BK	10
Tabel 2.5 Daftar bahan pakan sapi perah	11
Tabel 4.1 Total kebutuhan nutrisi sapi.....	28
Tabel 4.2 Populasi awal partikel	29
Tabel 4.3 Total ketersediaan nutrisi partikel ke-1	31
Tabel 4.4 Nilai <i>fitness</i> pada tiap partikel.....	33
Tabel 4.5 Posisi lokal terbaik pada tiap partikel	33
Tabel 4.6 Posisi global terbaik pada tiap partikel	34
Tabel 4.7 Kecepatan partikel terbaru	35
Tabel 4.8 Posisi partikel terbaru	35
Tabel 4.9 Rancangan pengujian jumlah iterasi	36
Tabel 4.10 Rancangan pengujian ukuran populasi	36
Tabel 4.11 Rancangan pengujian interval berat bahan pakan	37
Tabel 6.1 Hasil pengujian jumlah iterasi	50
Tabel 6.2 Hasil pengujian ukuran populasi	52
Tabel 6.3 Hasil pengujian interval berat bahan pakan	53
Tabel 6.4 hasil observasi di desa Dresel, kecamatan Batu, kota Batu.....	54
Tabel 6.5 Hasil dengan sistem optimasi komposisi pakan sapi perah menggunakan IPSO	55
Tabel 6.6 Hasil pengujian data	56
Tabel 6.7 Hasil optimasi bahan pakan sapi perah menggunakan algoritma PSO. 56	
Tabel 6.8 Hasil optimasi bahan pakan sapi perah menggunakan algoritma IPSO 57	
Tabel 6.9 Hasil pengujian perbandingan algoritma	57

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi pembaruan kecepatan dan posisi dari sebuah partikel pada ruang dua dimensi.....	13
Gambar 3.1 Metodologi penelitian.....	16
Gambar 3.2 Alur kerja sistem.....	18
Gambar 4.1 Diagram alir penyelesaian masalah menggunakan IPSO.....	21
Gambar 4.2 Diagram alir proses membangun populasi awal.....	23
Gambar 4.3 Diagram alir proses menghitung nilai <i>fitness</i>	24
Gambar 4.4 Diagram alir proses menentukan posisi lokal terbaik.....	25
Gambar 4.5 Diagram alir menentukan posisi global terbaik.....	25
Gambar 4.6 Diagram alir menentukan kecepatan partikel.....	26
Gambar 4.7 Diagram alir menentukan posisi partikel.....	26
Gambar 4.8 <i>Entity Relationship Diagram</i> (ERD).....	38
Gambar 4.9 Rancangan antarmuka <i>input</i> data.....	38
Gambar 4.10 Rancangan antarmuka pengelolaan data bahan pakan.....	39
Gambar 4.11 Rancangan antarmuka <i>input</i> dan <i>edit</i> data bahan pakan.....	40
Gambar 5.1 Implementasi basis data.....	46
Gambar 5.2 Implementasi antarmuka <i>input</i> data.....	47
Gambar 5.3 Implementasi antarmuka pengelolaan data bahan pakan.....	48
Gambar 5.4 Implementasi antarmuka <i>input</i> data bahan pakan.....	48
Gambar 5.5 Implementasi antarmuka <i>edit</i> data bahan pakan.....	49
Gambar 6.1 Grafik hasil pengujian jumlah iterasi.....	51
Gambar 6.2 Grafik hasil pengujian ukuran populasi.....	52
Gambar 6.3 Grafik hasil pengujian interval berat bahan pakan.....	54

DAFTAR LAMPIRAN

LAMPIRAN A Hasil Wawancara Narasumber	62
LAMPIRAN B Hasil Observasi Data Penerimaan Susu Di KUD Batu dan Data Pemberian Pakan Peternak Di Desa Dresel, Kecamatan Batu, Kota Batu	63



BAB 1 PENDAHULUAN

1.1 Latar belakang

Sistem pemberian pakan mempunyai peranan penting dalam usaha sapi perah. Ketersediaan pakan yang memadai dapat memaksimalkan produksi susu sehingga menunjang keberhasilan usaha sapi perah (Bamualin, et al., 2009) (Jayanegara, 2014). Supaya produksi susu yang diperoleh maksimal maka pakan yang disediakan harus mencukupi kebutuhan nutrisi sapi. Kebutuhan nutrisi sapi perah meliputi kebutuhan nutrisi untuk hidup pokok sapi dan kebutuhan nutrisi untuk produksi susu. Untuk mencukupi kebutuhan tersebut, peternak sapi dianjurkan untuk memberikan bahan pakan hijauan dan juga konsentrat. Pemberian bahan pakan hijauan dan konsentrat yang tepat tidak hanya dapat memenuhi kebutuhan hidup pokok sapi, melainkan juga dapat memaksimalkan potensi genetik sapi dalam memproduksi susu (Lampiran A).

Namun di sisi lain harga pakan merupakan masalah sendiri bagi peternak sapi. Dalam usaha sapi perah, biaya pakan memegang prosentase terbesar yang mencapai 62,5% dari total biaya usaha (Umiyasih & Anggraeny, 2008) (Yusdja, 2005). Pemberian pakan yang tidak tepat bisa mengakibatkan peternak mengeluarkan biaya yang besar hanya untuk pakan tetapi produksi susu yang dihasilkan tidak maksimal. Untuk itu diperlukan sebuah metode yang dapat menyusun bahan pakan sesuai kebutuhan nutrisi sapi dengan harga pakan yang seminimal mungkin. Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah kombinatorik semacam ini, diantaranya algoritma genetika, *evolution strategies* (ES), dan algoritma *improved particle swarm optimization* (IPSO).

Seperti penelitian Fakhroh, et al. (2015) yang membuat sistem penyusunan pakan sapi perah menggunakan algoritma genetika. Selain memperhatikan kondisi sapi, penyusunan pakan sapi juga didasarkan pada harga pakan. Jadi hasil akhir yang diharapkan adalah komposisi pakan sapi yang optimal dengan biaya yang serendah mungkin. Untuk mewujudkan keluaran yang diharapkan, Fakhroh, et al. (2015) menggunakan algoritma genetika. Metode tersebut digunakan karena dinilai dapat menyelesaikan masalah optimasi dengan tetap memperhatikan batasan yang ada. Dari penelitian tersebut diketahui bahwa algoritma genetika mampu menghasilkan keluaran komposisi pakan sapi yang optimal dengan biaya pakan yang minimum.

Algoritma lain yang diketahui dapat menyelesaikan permasalahan semacam ini adalah *evolution strategies* (ES). Dalam penelitian yang dilakukan oleh Milah & Mahmudy (2015), algoritma ES digunakan untuk menentukan komposisi pakan sapi potong dengan biaya pakan yang minimum. Berbeda dari penelitian sebelumnya, penelitian ini bertujuan untuk mendapatkan komposisi pakan sapi potong yang optimal sehingga dapat meningkatkan produktivitas sapi potong. Dari hasil pengujian disimpulkan bahwa algoritma ES mampu menghasilkan solusi yang optimal dengan biaya pakan yang rendah.

Selain algoritma genetika dan ES, algoritma *improved particle swarm optimization* (IPSO) juga diketahui dapat menyelesaikan masalah yang menyangkut optimasi. Seperti Wu, et al. (2015) yang menerapkan algoritma IPSO untuk mengatasi masalah keandalan. Dalam penelitiannya, Wu, et al. (2015) membandingkan algoritma IPSO dengan empat algoritma lain untuk menyelesaikan lima contoh masalah. Dari pengujian tersebut didapatkan bahwa algoritma IPSO terbukti lebih unggul dari dua algoritma lainnya.

Penelitian tentang algoritma IPSO juga dilakukan oleh Chen, et al. (2015). Pada penelitian tersebut, algoritma IPSO digunakan untuk optimasi distribusi sumber daya pada *hybrid electric vehicles* (HEVs). Chen, et al. (2015) mengungkapkan bahwa algoritma yang berbasis PSO dinilai lebih efisien dan membutuhkan tingkat komputasi yang lebih rendah dibandingkan algoritma genetika. Berdasarkan hasil penelitian yang didapatkan, algoritma IPSO terbukti lebih efisien dalam menemukan solusi optimum dibandingkan dengan metode PSO konvensional.

Algoritma IPSO juga digunakan oleh Lu, et al. (2015) untuk klasifikasi teks yang dikombinasikan dengan metode KNN *classifier*. Algoritma berbasis PSO dipilih karena lebih mudah diimplementasikan dan lebih efisien dalam mencari solusi optimum jika dibandingkan dengan algoritma semut dan algoritma genetika. Setelah dilakukan pengujian, didapatkan bahwa program IPSO yang menggunakan *inertia weight* dan *constriction* secara asinkron dapat menghasilkan nilai *fitness* akhir terbaik dan dapat konvergen dalam waktu yang relatif cepat (Lu, et al., 2015).

Seperti yang telah dijelaskan sebelumnya bahwa sistem pemberian pakan pada sapi perah berperan penting dalam produktivitas susu. Oleh karena itu, untuk meningkatkan efisiensi dalam penyusunan bahan pakan, penulis merancang sistem untuk optimasi komposisi pakan sapi perah menggunakan algoritma IPSO. Metode IPSO dipilih karena terbukti lebih unggul dari pada algoritma PSO konvensional dan memerlukan tingkat komputasi yang lebih kecil dibandingkan algoritma genetika (Chen, et al., 2015). Selain itu, algoritma berbasis PSO dinilai lebih mudah diimplementasikan dan lebih efisien dalam mencapai solusi optimum jika dibandingkan dengan algoritma semut dan algoritma genetika (Lu, et al., 2015).

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi algoritma IPSO untuk menentukan komposisi pakan sapi perah yang optimal?
2. Bagaimana menentukan parameter algoritma IPSO yang tepat supaya dapat menghasilkan komposisi pakan sapi perah yang optimal?
3. Bagaimana kualitas solusi yang dihasilkan menggunakan IPSO?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian adalah sebagai berikut:

1. Merancang dan mengimplementasikan algoritma IPSO untuk menentukan komposisi pakan sapi perah optimal.
2. Menentukan parameter algoritma IPSO yang tepat supaya dapat menghasilkan komposisi pakan sapi perah yang optimal.
3. Mengetahui kualitas solusi yang dihasilkan menggunakan algoritma IPSO.

1.4 Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk pihak – pihak berikut ini:

1. Bagi Penulis

Penulis dapat memahami tentang implementasi algoritma IPSO untuk menentukan komposisi pakan sapi perah yang sesuai dengan kebutuhan sapi.

2. Bagi Masyarakat

Mempermudah masyarakat terutama para peternak sapi dalam menentukan komposisi pakan sapi perah yang tepat sesuai kondisi sapi dengan harga pakan yang murah sehingga dapat memaksimalkan produksi susu.

3. Bagi Pemerintah

Diharapkan dapat membantu usaha pemerintah dalam meningkatkan produktivitas pada usaha sapi perah.

1.5 Batasan masalah

Untuk membatasi ruang lingkup pada penelitian ini maka ditetapkan batasan sebagai berikut:

1. Data yang digunakan dalam penelitian ini meliputi data pemberian bahan pakan yang dilakukan peternak sapi di kota Batu dan data penerimaan susu sapi yang diperoleh dari KUD Batu.
2. Ada 10 jenis bahan pakan yang akan dioptimasi dalam penelitian ini.
3. Proses optimasi bahan pakan sapi perah memerlukan masukan berat badan sapi, rata-rata produksi susu sapi, rata-rata kadar lemak susu, dan bahan pakan yang akan diberikan.
4. Keluaran dari proses optimasi adalah komposisi bahan pakan yang sesuai dengan kondisi sapi perah dengan harga pakan yang seminimal mungkin.

1.6 Sistematika pembahasan

Agar tujuan dapat tercapai maka penelitian ini disusun dengan sistematika sebagai berikut:

BAB I : Pendahuluan

Bab ini menjelaskan mengenai latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II : Landasan Kepustakaan

Bab ini membahas kajian pustaka dan teori – teori yang berkaitan dengan penyelesaian kasus penyusunan pakan sapi perah menggunakan algoritma IPSO. Kajian pustaka berisi beberapa penelitian sebelumnya yang berkaitan dengan obyek dan metode dari penelitian ini. Teori yang dibutuhkan pada penelitian ini antara lain: teori yang berkaitan dengan pakan sapi perah dan algoritma IPSO.

BAB III : Metodologi

Bab ini berisi langkah – langkah untuk menyelesaikan penelitian yang terdiri dari studi literatur, analisa kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian, dan pengambilan keputusan.

BAB IV : Perancangan

Bab ini berisi formulasi permasalahan, alur penyelesaian menggunakan algoritma IPSO, manualisasi permasalahan menggunakan algoritma IPSO, perancangan pengujian, perancangan basis data, dan perancangan antarmuka.

BAB V : Implementasi

Bab ini menjelaskan tentang implementasi sistem berdasarkan perancangan yang telah dibuat.

BAB VI : Pengujian dan Pembahasan

Bab ini berisi proses dan hasil pengujian terhadap metode yang digunakan serta analisis terhadap hasil pengujian yang didapatkan. Proses pengujian dilakukan berdasarkan perancangan pengujian.

BAB VII : Penutup

Bab ini memuat kesimpulan berdasarkan hasil pengujian yang didapatkan dan saran – saran untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka dan dasar teori yang digunakan untuk menunjang topik pembahasan. Kajian pustaka menjelaskan penelitian – penelitian sebelumnya yang memiliki korelasi dengan obyek dan metode pada penelitian ini. Sedangkan dasar teori menjelaskan mengenai beberapa teori yang dibutuhkan untuk penyusunan penelitian ini. Beberapa teori yang dibutuhkan adalah teori yang berkaitan dengan pakan sapi perah dan algoritma *improved particle swarm optimization* (IPSO).

2.1 Kajian Pustaka

Kajian pustaka berisi penjelasan penelitian – penelitian sebelumnya yang relevan dengan penelitian yang sedang dilakukan. Obyek pada penelitian ini adalah komposisi pakan sapi perah dengan metode IPSO. Maka penelitian yang dikaji merupakan penelitian yang menggunakan obyek atau permasalahan yang hampir sama dengan obyek yang sedang diteliti serta penelitian lain yang membahas tentang metode IPSO.

Kajian yang pertama merupakan penelitian yang dilakukan oleh Fakhroh, et al. (2015). Penelitian tersebut bertujuan untuk mendapatkan komposisi pakan sapi perah yang optimal dengan biaya pakan serendah mungkin. Untuk mencapai tujuan tersebut Fakhroh, et al. (2015) menerapkan algoritma genetika. Metode tersebut dipilih karena dianggap mampu menghasilkan keluaran yang optimal dengan tetap memperhatikan batasan yang ditentukan (Fakhroh, et al., 2015). Data yang digunakan meliputi daftar kebutuhan pokok hidup sapi, daftar kebutuhan untuk memproduksi satu liter susu, dan sejumlah daftar kandungan bahan pakan serta data mengenai harga bahan pakan sapi. Di akhir penelitian, Fakhroh, et al. (2015) menyimpulkan bahwa algoritma genetika mampu memaksimalkan komposisi pakan sapi perah dengan biaya pakan yang minimal.

Penelitian Milah & Mahmudy (2015) juga membahas tentang optimasi pakan sapi namun menggunakan metode yang berbeda yaitu *evolution strategies* (ES). Obyek yang diteliti juga bukan sapi perah melainkan sapi potong. Oleh karena itu penelitian ini bertujuan untuk mendapatkan komposisi pakan sapi potong yang optimal dengan biaya yang minimal supaya dapat meningkatkan produktivitas sapi potong. Metode ES digunakan karena menurut Mahmudy, et al. (2013) algoritma ES dapat digunakan untuk menyelesaikan masalah yang sulit dipecahkan dengan pendekatan analisis berbasis model matematis (Milah & Mahmudy, 2015). Dari pengujian yang dilakukan didapatkan bahwa ES tidak menghasilkan *penalty* sehingga ES dapat dikatakan sebagai algoritma yang dapat menyelesaikan masalah optimasi pakan sapi ternak dengan tetap memperhatikan harga pakan.

Tidak hanya algoritma genetika dan *Evolution Strategies* (ES) yang dapat menyelesaikan masalah optimasi komposisi pakan sapi. Algoritma *Improved Particle Swarm Optimization* (IPSO) juga dapat menyelesaikan persoalan semacam ini. Seperti Wu, et al. (2011) yang menerapkan algoritma IPSO untuk

menyelesaikan masalah keandalan. Dikatakan bahwa masalah keandalan merupakan masalah yang memperhatikan biaya, bobot, dan *volume* (Wu, et al., 2011). Sama halnya dengan obyek pada penelitian ini yang juga memperhatikan biaya dan bobot. Dalam penelitiannya, Wu, et al. (2011) menggunakan lima contoh masalah untuk menguji unjuk kerja dari algoritma IPSO. Kelima masalah tersebut diselesaikan dengan metode IPSO dan empat algoritma lain sebagai pembandingan. Empat algoritma itu antara lain *particle swarm optimization* (PSO), *particle swarm optimization* berdasarkan *inertia weight* (IWPSO), *bare bones particle swarm optimization* (BBPSO), dan *bare bones particle swarm optimization* dengan *gaussian or cauchy jump* (BBPSO+GJ/CJ). Dari pengujian yang telah dilakukan, didapatkan bahwa algoritma IPSO terbukti lebih efisien dalam mendapatkan solusi optimum dari pada algoritma PSO dan IWPSO.

Penerapan algoritma IPSO juga dilakukan Chen, et al. (2015) untuk memaksimalkan pengelolaan distribusi sumber daya pada *hybrid electric vehicles* (HEVs). Algoritma PSO dipilih karena dianggap cocok untuk menangani berbagai macam masalah (Chen, et al., 2015). Dibandingkan dengan algoritma genetika, algoritma PSO dinilai lebih efektif dalam mendapatkan solusi optimum karena dapat mengingat solusi optimum dari iterasi sebelumnya untuk digunakan pada iterasi selanjutnya. Chen, et al (2015) juga menambahkan bahwa algoritma PSO memerlukan tingkat komputasi yang rendah jika dibandingkan dengan algoritma genetika karena pada algoritma PSO tidak ada proses reproduksi, *crossover*, dan mutasi yang memerlukan tingkat komputasi yang tinggi. Hasil pengujian pada penelitian ini juga menyatakan bahwa algoritma IPSO dapat menghasilkan solusi optimum dengan lebih efisien dari pada algoritma PSO konvensional.

Satu lagi kajian pustaka yang menggunakan algoritma IPSO adalah penelitian yang dilakukan oleh Lu, et al. (2015). Penelitian ini membahas tentang penerapan algoritma IPSO yang dikombinasikan dengan metode KNN *classifier* untuk klasifikasi teks. Algoritma PSO dipilih karena lebih mudah diimplementasikan dan lebih efisien dalam mencari solusi optimum jika dibandingkan dengan algoritma semut dan algoritma genetika. Dalam penelitiannya, Lu, et al. (2015) menambahkan komponen *inertia weight* dan *constriction factor* pada algoritma PSO sehingga menjadi beberapa program. Pada program pertama *inertia weight* dibuat konstan 0,9, kemudian dikembangkan sehingga menjadi dinamis dan disebut sebagai program kedua. Setelah itu *inertia weight* diganti dengan *constriction factor* yang awalnya dibuat konstan 0,7298 dan disebut sebagai program tiga. Selanjutnya *constriction factor* dikembangkan menjadi fungsi kosinus dan disebut sebagai program empat. Kemudian program kelima menggabungkan *inertia weight* dan *constriction factor* secara sinkron atau bersamaan. Dan terakhir program ke enam, *inertia weight* dan *constriction factor* digunakan secara terpisah atau asinkron. Setelah dilakukan pengujian terhadap enam program tersebut, didapatkan bahwa program ke enam merupakan model yang terbaik diantara lima model lainnya. Dikatakan bahwa program ke enam menghasilkan nilai *fitness* akhir terbaik dan dapat konvergen dalam waktu yang relatif cepat (Lu, et al., 2015).

Berdasarkan kajian pustaka di atas, penulis menggunakan metode algoritma IPSO untuk menyelesaikan kasus optimasi komposisi pakan sapi perah. Algoritma IPSO yang digunakan pada penelitian ini merujuk pada penelitian yang dilakukan oleh Lu, et al. (2015). Penjelasan detail mengenai algoritma IPSO dijelaskan pada sub-bab selanjutnya. Sedangkan rangkuman dari penelitian yang dikaji dan perbedaannya dengan penelitian yang diajukan penulis disajikan pada Tabel 2.1.

Tabel 2.1 Kajian pustaka

No	Judul	Obyek	Metode	Hasil
1	Optimasi komposisi pakan sapi perah menggunakan algoritma genetika (Fakhiroh, et al., 2015)	Komposisi pakan sapi perah	Algoritma genetika	Penerapan algoritma genetika pada penelitian ini dikatakan mampu memaksimalkan komposisi pakan sapi perah dengan biaya pakan yang minimal
2	Implementasi algoritma evolution strategies untuk optimasi komposisi pakan ternak sapi potong (Milah & Mahmudy, 2015)	Komposisi pakan ternak sapi potong	<i>Evolution strategies</i> (ES)	Dari hasil pengujian menunjukkan jika ES tidak menghasilkan <i>penalty</i> sehingga ES dapat dikatakan sebagai algoritma yang dapat menyelesaikan masalah optimasi pakan sapi ternak dengan tetap memperhatikan harga pakan
3	An improved particle swarm optimization algorithm for reliability problems (Wu, et al., 2011)	Masalah keandalan	<i>Improved particle swarm optimization</i> (IPSO)	Setelah proses pengujian dengan empat algoritma lain terbukti bahwa algoritma IPSO lebih efisien dalam mendapatkan solusi optimum daripada algoritma PSO dan IWPSO
4	Optimal energy management of a hybrid electric powertrain system (Chen, et al., 2015)	Distribusi sumber daya pada <i>hybrid electric vehicles</i> (HEVs)	<i>Improved particle swarm optimization</i> (IPSO)	Berdasarkan hasil pengujian didapatkan bahwa algoritma IPSO mampu menghasilkan solusi optimum dengan lebih efisien daripada algoritma PSO konvensional
5	Improved particle swarm optimization algorithm and its application in text	Klasifikasi teks	<i>Improved particle swarm optimization</i> (IPSO)	Dari hasil pengujian didapatkan bahwa program yang menerapkan <i>inertia weight</i> dan <i>constriction</i>

No	Judul	Obyek	Metode	Hasil
	feature selection (Lu, et al., 2015)			<i>factor</i> secara asinkron menghasilkan nilai <i>fitness</i> akhir terbaik dan dapat konvergen dalam waktu yang relatif cepat
6	Usulan Penulis : Optimasi komposisi pakan sapi perah menggunakan improved particle swarm optimization (IPSO)	Komposisi pakan sapi perah	<i>Improved particle swarm optimization (IPSO)</i>	Penggunaan algoritma IPSO yang diajukan dalam penelitian ini mampu menghasilkan komposisi pakan sapi perah yang optimum sehingga dapat memenuhi kebutuhan nutrisi pada sapi serta memaksimalkan produktivitas susu dengan harga pakan yang serendah mungkin.

2.2 Pakan Sapi Perah

Sapi perah merupakan sapi yang dikembangkan dengan tujuan untuk diambil air susunya. Secara umum, sapi yang ada di dunia ini terbagi menjadi dua jenis yaitu sapi yang berasal dari daerah tropis (*Bos Indicus*) dan sapi yang berasal dari daerah sub tropis (*Bos Taurus*). Salah satu sapi yang termasuk dalam *Bos Indicus* adalah sapi *Friesen Holstein* (FH) (Ako, 2011). Menurut Ditjennak (2010), Sapi FH merupakan jenis sapi perah yang banyak dipelihara di dunia termasuk di Indonesia. Hampir 495.089 ekor sapi FH dipelihara di Indonesia dan memenuhi 35% kebutuhan susu nasional (Aggraeni, 2012).

Sapi perah akan memproduksi susu dalam jumlah tinggi ketika dalam masa laktasi yang artinya masa produksi menghasilkan susu. Seperti yang telah dijelaskan sebelumnya, komposisi bahan pakan mempunyai peran penting dalam produktivitas susu. Penyusunan bahan pakan yang tepat dapat memaksimalkan produksi susu. Bahan pakan merupakan bahan yang dapat dimakan, dicerna dan dimanfaatkan oleh hewan sebagai sumber energi dalam tubuhnya. Ada dua jenis bahan makanan yang dibutuhkan oleh sapi perah antara lain sebagai berikut (Fakhiroh, et al., 2015) (Umiyasih & Anggraeny, 2008).

- Makanan kasar

Makanan kasar adalah bahan pakan yang memiliki tingkat serat yang tinggi. Bahan ini biasanya disusun dari bahan hijauan seperti rumput atau leguminose yang masih segar.

- Makanan penguat (konsentrat)

Makanan penguat merupakan kebalikan dari makanan kasar atau bahan pakan yang memiliki tingkat serat yang rendah. Makanan ini berfungsi sebagai penyeimbang makanan kasar karena sifatnya yang mudah dicerna. Selain itu

makanan penguat juga berfungsi memenuhi zat – zat makanan yang tidak terdapat dalam makanan kasar. Konsentrat biasanya disusun dari limbah pertanian dan agroindustry seperti dedak padi, bungkil, dan onggok. Bahan lainnya yang bisa digunakan sebagai konsentrat antara lain ampas tahu, ampas kecap, dan molasses.

Untuk mendapatkan produksi susu yang optimal maka proporsi antara hijauan (makanan kasar) dan konsentrat harus seimbang. Susunan bahan makanan yang terdiri dari satu atau lebih bahan pakan dan diberikan kepada seekor ternak pada sehari semalam disebut dengan ransum. Pemberian ransum pada seekor sapi perah harus memenuhi nutrisi yang dibutuhkan untuk hidup pokok sapi, pertumbuhan, dan produksi susu (Alfian, 2014). Jumlah nutrisi yang dibutuhkan seekor sapi perah didasarkan pada berat badan sapi, jumlah susu yang diproduksi, dan kandungan lemak pada susu tersebut (lampiran A).

Ada tiga nutrisi yang biasanya terkandung dalam bahan pakan sapi perah antara lain bahan kering (BK), protein kasar (PK), dan *Total Digestible Nutrient* (TDN). Nutrisi inilah yang nantinya akan dioptimasi dalam penyusunan bahan pakan sapi perah. Selain dari tiga nutrisi tersebut, seekor sapi perah juga membutuhkan *Net Energy of Lactation* (NE_L) untuk memproduksi susu (Alfian, 2014). Tabel 2.2 menunjukkan kebutuhan nutrisi untuk hidup pokok sapi.

Tabel 2.2 Kebutuhan nutrisi untuk hidup pokok sapi perah laktasi

BB (Kg)	NE _L (Mkal)	TDN (Kg)	PK (g)
350	6,47	2,85	341
400	7,16	3,15	373
450	7,82	3,44	403
500	8,46	3,72	432
550	9,09	4,00	461

Sumber: (Alfian, 2014)

Selain untuk kebutuhan hidup pokok sapi, bahan pakan yang diberikan pada sapi perah juga harus memenuhi kebutuhan nutrisi untuk produksi susu. Kebutuhan nutrisi untuk menghasilkan satu liter susu dijelaskan melalui Tabel 2.3.

Tabel 2.3 Kebutuhan nutrisi untuk produksi 1 liter susu

Kadar Lemak (%)	NE _L (Mkal)	TDN (Kg)	PK (g)
2,5	0,59	0,260	72
3,0	0,64	0,282	77
3,5	0,69	0,364	82
4,0	0,74	0,326	87
4,5	0,78	0,344	92
5,0	0,83	0,365	98
5,5	0,88	0,387	103
6,0	0,93	0,410	108

Sumber: (Alfian, 2014)

Jika kondisi berat badan sapi tidak ditemukan pada Tabel 2.2 maka digunakan persamaan 2-1 berikut. Persamaan 2-1 juga berlaku untuk kadar lemak susu yang tidak tersedia pada Tabel 2.3.

$$y = \frac{y_{atas} - y_{bawah}}{x_{atas} - x_{bawah}} \times (x - x_{bawah}) + y_{bawah} \quad (2-1)$$

Dengan keterangan,

- y = nilai nutrisi yang didapatkan
- x = nilai *input* (berat badan sapi atau kadar lemak susu)
- x_{bawah} = nilai yang tepat di bawah x
- x_{atas} = nilai yang tepat di atas x
- y_{bawah} = nilai nutrisi dari x_{bawah}
- y_{atas} = nilai nutrisi dari x_{atas}

Sebagai contoh terdapat berat badan sapi (BB) sebesar 370. Pada Tabel 2.2, nilai BB 370 terletak diantara BB 350 dan BB 400. Nilai 350 akan menjadi x_{bawah} dan nilai 400 akan menjadi x_{atas}. Untuk mendapatkan nutrisi NE_L dengan BB 370 maka perhitungannya sebagai berikut.

$$y = \frac{NE_L \text{ dari } x_{atas} - NE_L \text{ dari } x_{bawah}}{x_{atas} - x_{bawah}} \times (x - x_{bawah}) + NE_L \text{ dari } x_{bawah}$$

$$y = \frac{7,16 - 6,47}{400 - 350} \times (370 - 350) + 6,47 = 6,746$$

Untuk mendapatkan nutrisi TDN dan PK, nilai NE_L diganti dengan TDN atau PK. Tabel 2.2 dan Tabel 2.3 digunakan untuk mendapatkan informasi tentang nutrisi NE_L, TDN, dan PK. Sedangkan untuk kebutuhan nutrisi BK ditentukan berdasarkan Tabel 2.4.

Tabel 2.4 Kebutuhan NE_L per kg BK

Produksi susu (L)	NEL/Kg BK (Mkal/Kg)
< 8	1,42
8 - 12	1,52
13 - 18	1,62
> 18	1,72

Sumber: (Alfian, 2014)

Kebutuhan nutrisi sapi perah yang juga harus dipenuhi adalah kebutuhan nutrisi untuk pertumbuhan. Kebutuhan ini sebesar sepuluh persen dari total NE_L pada kebutuhan hidup pokok sapi seperti pada persamaan 2-2 berikut (Alfian, 2014).

$$\text{Pertumbuhan (Mkal)} = 10\% \times NE_L \text{ hidup pokok} \quad (2-2)$$

Setiap jenis bahan pakan mempunyai kandungan nutrisi yang berbeda. Pada penelitian ini ada sepuluh jenis bahan pakan sapi yang akan digunakan untuk optimasi komposisi pakan sapi perah. Tabel 2.4 menampilkan jenis bahan pakan yang digunakan beserta kandungan gizi yang dimiliki dan juga harga per kilogram nya.

Tabel 2.5 Daftar bahan pakan sapi perah

Bahan Pakan	BK (%)	PK (%)	TDN (%)	Harga (Rp/Kg)
Rumput gajah	36,5	12,8	55,7	1250
Tebon jagung	35,1	8,8	68,8	1700
Pollard	82,82	18,12	84,78	4500
Bapro	81,79	18,69	32,456	3000
Ampas Tahu	20,66	25,69	73,625	500
Limbah roti	68,3	15	89,3	250
Jerami padi	80,8	3,9	45,47	500
Kulit kedelai	90,9	13,9	67,3	250
Daun Lamtoro	30	32,12	69,017	2000
Bekatul	90,6	15,5	84,8	2500

Sumber: (Alfian, 2014) (Council, 2001)

NE_L merupakan representasi dari energi yang terkandung dalam bahan pakan sapi perah laktasi. Sedangkan TDN merupakan data terkait energi yang tersedia pada suatu bahan pakan (Jayanegara, 2014). Oleh karena itu untuk mendapatkan NE_L, TDN perlu dikonversi menggunakan persamaan berikut (Council, 2001).

$$NE_L (Mkal/Kg) = 0.0245 \times TDN(\%) - 0.12 \quad (2-3)$$

2.3 Algoritma Improved Particle Swarm Optimization (IPSO)

Sub – bab ini menjelaskan tentang algoritma IPSO yang digunakan pada penelitian ini. Algoritma IPSO merupakan pengembangan dari algoritma Particle Swarm Optimization (PSO). Oleh karena itu penjelasan algoritma IPSO didahului dengan penjelasan mengenai algoritma PSO.

2.3.1 Algoritma Particle Swarm Optimization (PSO)

Algoritma PSO merupakan bagian dari *computational swarm intelligence* (CSI), yang mengacu pada perilaku sekelompok agen dalam menyelesaikan masalah dengan cara berkomunikasi dengan lingkungannya. CSI merupakan model yang dapat digunakan untuk menyelesaikan masalah yang kompleks terutama masalah yang menyangkut optimasi. Hal ini yang juga menjadi alasan penulis menggunakan algoritma improved PSO. Algoritma PSO pertama kali diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995, yang terinspirasi dari perilaku kawanan burung dalam menentukan pola supaya dapat terbang secara serentak dan membentuk grup dalam formasi yang optimal (Engelbrecht, 2007).

Setiap burung pada kelompok tersebut dianggap sebagai partikel yang merepresentasikan solusi dalam ruang pencarian d . Dengan kata lain PSO dapat diartikan sebagai kumpulan partikel dalam *swarm* yang bergerak dalam ruang pencarian. Tiap partikel yang ada di dalam *swarm* mempunyai atribut posisi ($x_{id}(t)$), kecepatan ($v_{id}(t)$), posisi lokal terbaik ($pbest_{id}$), dan posisi global terbaik ($gbest_d$). Istilah *swarm* di sini dapat diartikan sebagai populasi yang jumlahnya ditentukan secara acak pada kisaran nilai 20 sampai 50 (Tuegeh, et al., 2009). Untuk mendapatkan solusi yang optimum, proses algoritma PSO dilakukan dalam beberapa iterasi (T_{max}). Pada tiap iterasi (t), partikel bergerak dengan kecepatan yang dinamis untuk mendekati area penelurusan yang lebih baik atau solusi optimum. Untuk itu partikel menggunakan pengalamannya dan informasi dari partikel lain supaya dapat menuju solusi optimum. Kecepatan partikel terbaru diformulasikan sebagai persamaan 2-4 (Engelbrecht, 2007).

$$v_{id}(t+1) = v_{id}(t) + c_1 \times r_1(t) \times (pbest_{id}(t) - x_{id}(t)) + c_2 \times r_2(t) \times (gbest_d(t) - x_{id}(t)) \quad (2-4)$$

Setelah kecepatan partikel terbaru didapatkan selanjutnya posisi partikel diperbarui menggunakan persamaan 2-5 (Engelbrecht, 2007).

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2-5)$$

Berdasarkan persamaan 2-4, kecepatan partikel terbaru ($v_{id}(t+1)$) dipengaruhi oleh tiga komponen berikut (Engelbrecht, 2007).

- Komponen kecepatan sebelumnya

Melalui komponen ini, partikel dapat menyimpan kecepatan sebelumnya. Partikel menyimpan kecepatan sebelumnya supaya tidak mengalami perubahan kecepatan yang drastis pada iterasi selanjutnya. Komponen ini juga dikenal sebagai komponen inersia.

- Komponen kognitif

Komponen kognitif berperan sebagai memori pada tiap partikel yang menyimpan posisi terbaik yang pernah dilalui partikel tersebut. Komponen ini kemudian disebut sebagai posisi lokal terbaik ($pbest_{id}$). Dengan adanya komponen ini mengakibatkan partikel dapat kembali ke tempat terbaik yang pernah dilaluinya.

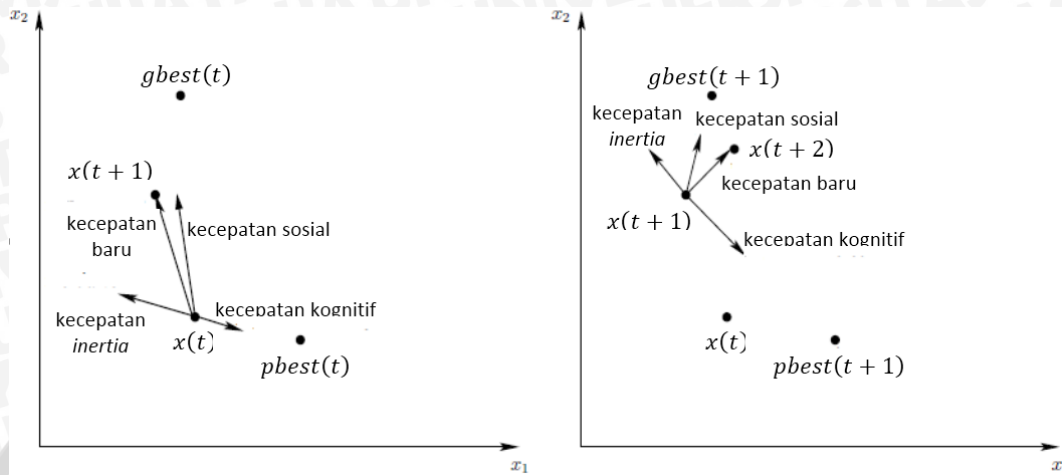
- Komponen sosial

Komponen sosial merujuk pada posisi terbaik yang pernah dicapai oleh suatu partikel dalam *swarm* atau yang kemudian disebut dengan posisi global terbaik ($gbest_d$). Komponen ini juga dapat mengakibatkan partikel kembali ke posisi terbaik yang pernah dicapai oleh partikel dalam *swarm*.

Seperti yang terlihat pada persamaan 2-4, komponen kognitif dipengaruhi oleh $c_1 r_1(t)$, begitu juga dengan komponen sosial yang dipengaruhi oleh $c_2 r_2(t)$. c_1 maupun c_2 merupakan koefisien akselerasi yang ditentukan secara acak pada rentang nilai 0 sampai 4, namun biasanya c_1 dan c_2 dibuat konstan dengan nilai

sama dengan 2 (Tuegeh, et al., 2009). Sedangkan $r_1(t)$ dan $r_2(t)$ merupakan angka antara 0 dan 1 yang dibangkitkan secara acak pada tiap iterasi (Pan & Liu, 2011).

Ilustrasi dari proses pembaruan kecepatan dan posisi sebuah partikel dapat dilihat melalui Gambar 2.1.



Gambar 2.1 Ilustrasi pembaruan kecepatan dan posisi dari sebuah partikel pada ruang dua dimensi

Sumber : (Engelbrecht, 2007)

Pada algoritma PSO juga terdapat fungsi *fitness* yang digunakan untuk mengukur kualitas dari partikel. Semakin baik nilai *fitness* suatu partikel maka semakin dekat pula partikel tersebut dengan solusi yang optimum. Fungsi *fitness* dari suatu masalah akan berbeda dengan fungsi *fitness* dari masalah yang lain sehingga fungsi ini ditentukan berdasarkan permasalahan yang dihadapi.

2.3.2 Algoritma *Improved Particle Swarm Optimization* (IPSO)

Improved Particle Swarm Optimization (IPSO) merupakan pengembangan dari algoritma PSO. Sebenarnya ada bermacam-macam model pengembangan algoritma PSO namun pada penelitian ini digunakan algoritma IPSO yang merujuk dari hasil penelitian Lu, et al. (2015). Pada algoritma IPSO ini diperkenalkan dua komponen yang sebelumnya tidak ada pada algoritma PSO konvensional. Komponen tersebut ialah *inertia weight* (w) dan *constriction factor* (K).

Lu, et al. (2015) menjelaskan bahwa penggunaan komponen *inertia weight* dan *constriction factor* secara asinkron dapat menghasilkan nilai *fitness* akhir yang tinggi dengan waktu konvergensi yang relatif singkat. *Inertia weight* diperkenalkan pertama kali oleh Shi dan Eberhart pada tahun 1998. Komponen ini digunakan sebagai penyeimbang antara penelusuran lokal dan global. Sedangkan *constriction factor* pertama kali diperkenalkan oleh Clerc yang fungsinya untuk memastikan konvergensi partikel (Lu, et al., 2015).

Di awal iterasi, partikel akan mencari solusi optimum pada area yang luas sehingga dibutuhkan nilai *constriction factor* yang besar. Kemudian semakin lama area penelusuran akan semakin kecil sehingga dibutuhkan nilai *constriction factor*

yang lebih kecil. Hal ini supaya partikel dapat mencapai solusi optimum secara lokal. Dari penjelasan tersebut maka *constriction factor* diformulasikan menjadi persamaan 2-6 berikut (Lu, et al., 2015).

$$K = \frac{\cos\left(\frac{2\pi}{T_{max}} \times \left(t - \frac{T_{max}}{2}\right)\right) + 2.428571}{4} \quad (2-6)$$

Dimana K merupakan *constriction factor*, T_{max} adalah jumlah iterasi maksimum, dan t adalah jumlah iterasi. Kemudian untuk formula *inertia weight* dikembangkan menjadi persamaan 2-7 berikut (Lu, et al., 2015).

$$w = \begin{cases} 0.857143 + (1 - 0.857143) \left(1 - \frac{t}{T_{max}}\right) & , gbest_d \neq x_{id} \\ 0.857143 & , gbest_d = x_{id} \end{cases} \quad (2-7)$$

Seperti yang telah dijelaskan di awal bahwa komponen *inertia weight* dan *constriction factor* digunakan secara asinkron atau tidak bersamaan. Hal ini dikarenakan perbedaan karakteristik dua komponen tersebut. Komponen *inertia weight* berpengaruh terhadap kecepatan orisinil partikel. Sedangkan komponen *constriction factor* berpengaruh terhadap konvergensi partikel. Oleh karena itu, dua komponen tersebut digunakan dalam periode yang berbeda. *Inertia weight* akan digunakan di setengah iterasi awal dengan tujuan untuk menyeimbangkan penelurusan global dan lokal. Kemudian *constriction factor* akan digunakan di setengah iterasi akhir untuk memastikan konvergensi partikel. Berdasarkan penjelasan tersebut maka formula kecepatan partikel pada persamaan 2-4 diperbarui menjadi persamaan 2-8 berikut (Lu, et al., 2015).

$$v_{id} = \begin{cases} w \times v_{id} + 2 \times r_1(pbest_{id} - x_{id}) + 2 \times r_2(gbest_d - x_{id}) & , t < \frac{T_{max}}{2} \\ K[0.7v_{id} + 2 \times r_1(pbest_{id} - x_{id}) + 2 \times r_2(gbest_d - x_{id})] & , t \geq \frac{T_{max}}{2} \end{cases} \quad (2-8)$$

Setelah didapatkan kecepatan partikel terbaru maka selanjutnya adalah memperbarui posisi partikel menggunakan persamaan 2-5. Adapun langkah – langkah dari algoritma IPSO adalah sebagai berikut (Chen, et al., 2015).

1. Inisialisasi

Inisialisasi merupakan proses inisialisasi posisi x_{ij} dimana $i = 1, 2, \dots, m$ merupakan jumlah partikel dan $j = 1, 2, \dots, n$ merupakan jumlah dimensi. Posisi x_{ij} ditentukan secara acak pada rentang nilai x_{min} sampai x_{max} .

2. Perhitungan nilai *fitness*

Perhitungan nilai *fitness* digunakan untuk mengukur kualitas solusi pada tiap partikel. Fungsi *fitness* ditentukan berdasarkan permasalahan yang dihadapi. Semakin baik nilai *fitness* pada suatu partikel maka solusi yang dihasilkan semakin mendekati optimum.

3. Seleksi dan penyimpanan

Tiap partikel menyimpan posisinya (x_{ij}) yang memiliki nilai *fitness* maksimum terbaik sebagai $pbest_{ij}$. Kemudian nilai *fitness* maksimum terbaik diantara semua partikel disimpan sebagai $gbest_j$.

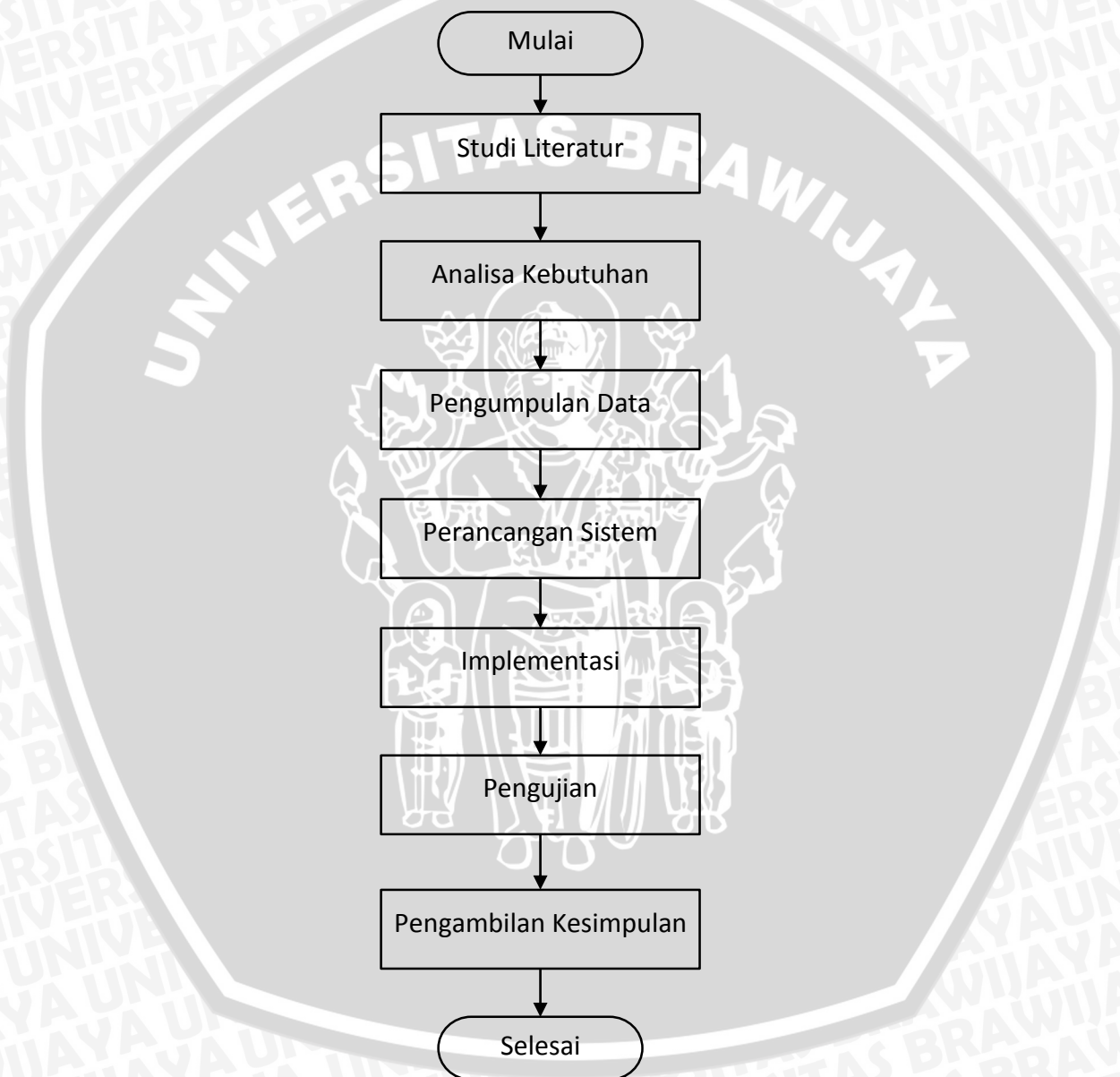
4. *Update* kecepatan dan posisi

Proses *update* kecepatan dilakukan berdasarkan persamaan 2-7. Setelah kecepatan terbaru didapatkan maka posisi partikel diperbarui menggunakan persamaan 2-4.



BAB 3 METODOLOGI

Bab ini menjelaskan metodologi atau langkah – langkah yang ditempuh untuk menyelesaikan masalah yang ada dalam penelitian ini. Adapun tahapan penyelesaiannya terdiri dari studi literatur, analisa kebutuhan, pengumpulan data, perancangan sistem, implementasi, pengujian, dan pengambilan kesimpulan. Gambar 3.1 menjelaskan alur dari metodologi yang diterapkan pada penelitian ini.



Gambar 3.1 Metodologi penelitian

3.1 Studi Literatur

Studi literatur digunakan untuk mendapatkan dasar teori dan kajian pustaka yang berfungsi sebagai acuan dalam penulisan penelitian dan pengembangan sistem. Kajian pustaka merupakan penjelasan penelitian – penelitian sebelumnya yang berkaitan dengan obyek dan metode yang digunakan dalam penelitian. Sedangkan dasar teori berisi kumpulan teori yang berkaitan dengan implementasi algoritma IPSO untuk optimasi pakan sapi perah. Teori – teori yang dipelajari antara lain teori tentang pakan sapi perah dan algoritma IPSO. Sumber literatur yang digunakan pada penelitian ini berasal dari jurnal, *conference proceeding*, buku, dan situs internet.

3.2 Analisa Kebutuhan

Analisa kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan dalam pengembangan sistem. Secara keseluruhan, kebutuhan yang digunakan untuk pembuatan sistem dalam penelitian ini meliputi:

- a. Kebutuhan Perangkat Keras
 - Komputer / *Laptop*.
- b. Kebutuhan Perangkat Lunak
 - *Microsoft Windows 7* sebagai sistem operasi.
 - *Microsoft Visio 2013* sebagai aplikasi untuk membuat diagram perancangan sistem.
 - *Netbeans 8.0.2* sebagai editor untuk membangun sistem berbasis web menggunakan *framework CodeIgniter* dan bahasa pemrograman PHP, HTML, CSS, dan *JavaScript*.
 - *Web Server XAMPP*.
 - Basis data MySQL.

3.3 Pengumpulan Data

Proses pengumpulan data meliputi mencari, memilih, dan mengumpulkan data yang diperlukan dalam penelitian. Data yang digunakan pada penelitian ini meliputi data pemberian pakan yang dilakukan oleh peternak sapi perah di desa Dresel, kabupaten Batu, kota Batu dan data penerimaan susu sapi di KUD kota Batu. Selain itu data juga didapatkan dari studi literatur berupa jurnal atau lainnya yang membahas tentang pakan untuk sapi perah. Data ini nantinya digunakan sebagai dasar untuk menghitung jumlah kebutuhan nutrisi sapi dan jumlah ketersediaan pakan yang akhirnya akan menghasilkan nilai *fitness*.

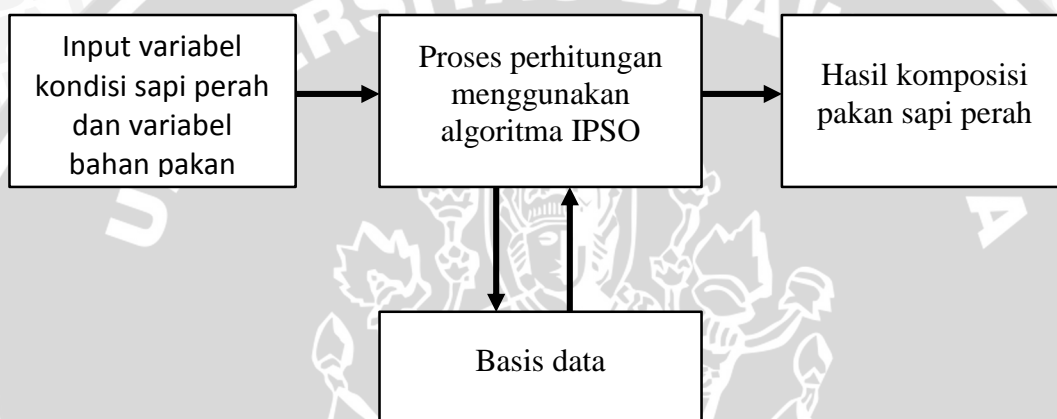
3.4 Perancangan Sistem

Perancangan sistem dilakukan untuk mentransformasikan kebutuhan yang telah ditetapkan sehingga dapat mempermudah proses implementasi dan

pengujian. Perancangan sistem yang dilakukan meliputi alur kerja sistem, pengujian, dan antarmuka. Berikut uraian perancangan yang akan disusun:

a. Alur Kerja Sistem

Alur kerja sistem menjelaskan jalannya sistem dalam menghasilkan solusi. Proses diawali dengan menginputkan variabel kondisi sapi dan variabel bahan pakan. Kemudian proses perhitungan menggunakan algoritma IPSO. Pada proses perhitungan, sistem akan mengakses basis data untuk mendapatkan informasi nutrisi bahan pakan yang dipilih. Basis data ini juga digunakan untuk menyimpan data bahan pakan yang dimasukkan oleh *user*. Dari data bahan yang ada, *user* juga dapat melakukan perubahan pada data tersebut. Dan terakhir proses menampilkan hasil perhitungan berupa komposisi pakan sapi perah yang mendekati optimum. Alur kerja sistem juga dapat dilihat melalui Gambar 3.1 berikut.



Gambar 3.2 Alur kerja sistem

b. Perancangan Pengujian

Perancangan pengujian merupakan rancangan untuk menguji parameter IPSO yang meliputi:

- Pengujian jumlah iterasi.
- Pengujian ukuran populasi partikel.
- Pengujian interval berat bahan pakan.

c. Perancangan Antarmuka

Antarmuka digunakan sebagai media komunikasi antara pengguna dengan sistem. Untuk itu perancangan antarmuka dilakukan untuk menyusun antarmuka yang dapat mempermudah komunikasi antara pengguna dan sistem.

3.5 Implementasi

Implementasi merupakan proses menyusun baris perintah menggunakan aplikasi editor dan bahasa pemrograman yang telah ditentukan. Proses implementasi dilakukan berdasarkan hasil perancangan yang telah dibuat. Implementasi yang dilakukan meliputi implementasi algoritma, basis data dan antarmuka. Keluaran yang dihasilkan dari implementasi sistem ini adalah komposisi pakan sapi perah optimal sesuai kondisi sapi dengan harga pakan yang minimal.

3.6 Pengujian

Pengujian dilakukan setelah proses implementasi selesai. Proses pengujian terdiri dari pengujian parameter IPSO, pengujian data, dan pengujian perbandingan algoritma. Pengujian parameter IPSO dilakukan berdasarkan perancangan pengujian yang telah ditetapkan. Pengujian parameter IPSO bertujuan untuk mendapatkan nilai optimum pada tiap parameter yang diuji coba. Dengan nilai yang optimum pada tiap parameter diharapkan dapat menghasilkan keluaran yang optimum pula. Pengujian data dilakukan dengan membandingkan hasil keluaran menggunakan sistem dengan data pemberian pakan para peternak di desa Dresel, kabupaten Batu, kota Batu. Sedangkan pengujian perbandingan algoritma dilakukan dengan membandingkan hasil optimasi menggunakan algoritma PSO konvensional dengan hasil optimasi menggunakan algoritma IPSO.

3.7 Pengambilan Kesimpulan

Pengambilan kesimpulan dapat dilakukan setelah proses perancangan, implementasi, dan pengujian selesai dikerjakan. Kesimpulan ditulis berdasarkan hasil pengujian yang telah didapatkan pada tahap sebelumnya. Penulisan kesimpulan diletakkan pada akhir penelitian beserta kekurangan dari penelitian ini dan saran untuk penelitian selanjutnya.



BAB 4 PERANCANGAN

Bab ini menjelaskan formulasi permasalahan, alur penyelesaian optimasi komposisi pakan sapi perah menggunakan algoritma *improved particle swarm optimization* (IPSO), perancangan pengujian, perancangan basis data dan perancangan antar muka.

4.1 Formulasi Permasalahan

Masalah yang diselesaikan adalah penyusunan bahan pakan yang tepat untuk sapi perah. Dalam menyelesaikan permasalahan ini sistem membutuhkan beberapa variabel diantaranya berat badan sapi, jumlah produksi susu, kadar lemak susu serta bahan pakan yang akan diberikan oleh *user* kepada sapi. Variabel berat badan sapi, produksi susu, dan kadar lemak digunakan untuk menghitung kebutuhan nutrisi untuk hidup pokok sapi dan kebutuhan nutrisi untuk produksi susu. Jumlah nutrisi yang dibutuhkan oleh sapi disebut sebagai total kebutuhan.

Setelah itu menghitung nutrisi dari bahan pakan yang telah dipilih. Hasil dari perhitungan nutrisi bahan pakan disebut sebagai total ketersediaan. Kemudian menghitung selisih antara total kebutuhan dan total ketersediaan sehingga membentuk total *penalty*. Jika nutrisi yang tersedia lebih kecil dari yang dibutuhkan maka *penalty* dihitung dengan cara total kebutuhan dikurangi total ketersediaan. Akan tetapi jika sebaliknya maka nilai *penalty* sama dengan total ketersediaan dikurangi total kebutuhan lalu dikali bobot *penalty*. Kemudian proses dilanjutkan dengan menghitung total harga dari bahan pakan yang dipilih. Setelah itu mengimplementasikan fungsi *fitness* yang melibatkan total *penalty* dan total harga pakan sehingga solusi yang dihasilkan adalah komposisi pakan yang mencukupi kebutuhan nutrisi sapi dengan harga pakan yang minimal.

Adapun contoh permasalahan yang akan diformulasikan dalam penelitian ini adalah sebagai berikut,

Terdapat seekor sapi perah dengan berat badan 450 kg, rata-rata produksi susu yang dihasilkan mencapai 15 L dengan rata-rata kadar lemak susu sebesar 4,5 %. Bahan pakan yang akan diberikan *user* diantaranya rumput gajah, tebon jagung, dan bapro. Untuk menyelesaikan persoalan ini, ukuran populasi yang digunakan adalah 5, jumlah iterasi maksimum sebanyak 2, dan batas berat maksimum pada tiap bahan pakan adalah 20 kg.

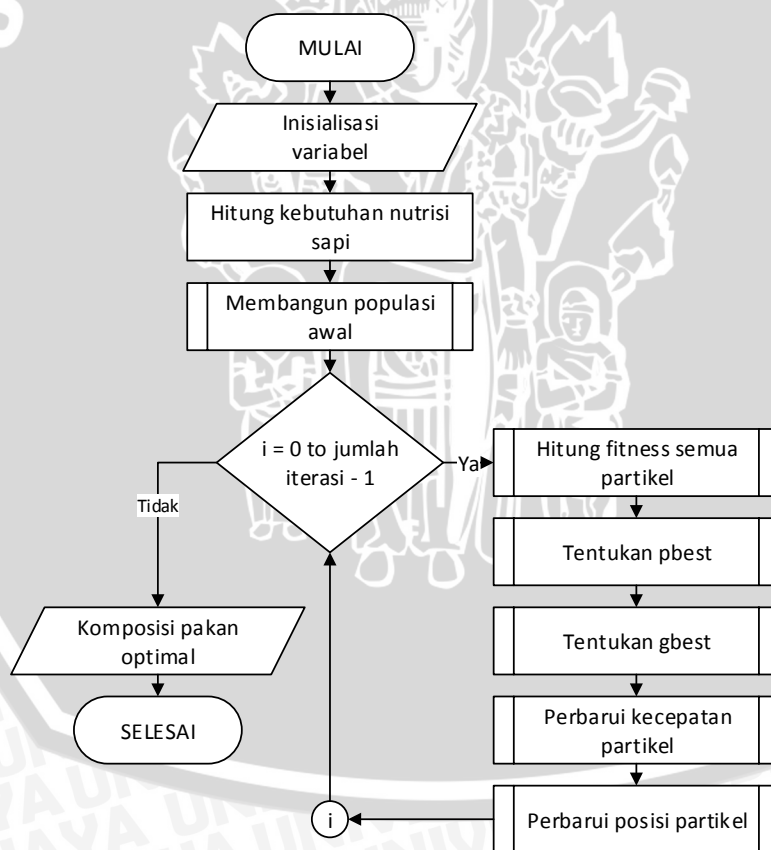
4.1.1 Menghitung Kebutuhan Nutrisi Sapi

Proses perhitungan kebutuhan nutrisi sapi berdasarkan pada variabel berat badan sapi, rata-rata produksi susu, dan kadar lemak susu. Kebutuhan nutrisi sapi perah meliputi kebutuhan nutrisi untuk hidup pokok dan kebutuhan nutrisi untuk produksi susu. Nutrisi yang harus dipenuhi untuk mencukupi kebutuhan tersebut antara lain bahan kering (BK), protein kasar (PK), *total digestible nutrient* (TDN), dan *net energy of lactation* (NE_L). Kebutuhan nutrisi untuk hidup pokok sapi ditentukan berdasarkan Tabel 2.2. Sedangkan kebutuhan nutrisi untuk produksi

susu dihitung berdasarkan Tabel 2.3. Tiap nutrisi yang didapatkan dari Tabel 2.3 dikalikan dengan variabel rata-rata produksi susu. Khusus nutrisi NE_L terdapat kebutuhan nutrisi untuk pertumbuhan yang dihitung menggunakan persamaan 2-1. Kemudian masing-masing nutrisi dari hidup pokok, produksi susu, dan pertumbuhan dijumlahkan sehingga membentuk total PK, total TDN, dan total NE_L . Setelah didapatkan total NE_L maka dapat ditentukan nutrisi BK yang dibutuhkan menggunakan Tabel 2.4. Total BK dihitung dengan membagi total NE_L dengan nilai yang didapatkan dari Tabel 2.4. Setelah empat nutrisi ini didapatkan maka perhitungan kebutuhan nutrisi sapi telah selesai dan didapatkan hasil akhir yang disebut dengan total kebutuhan.

4.2 Alur Penyelesaian Masalah Menggunakan Algoritma IPSO

Sub bab ini menjelaskan alur dari algoritma IPSO dalam menyelesaikan masalah. Proses diawali dengan inialisasi variabel, lalu dilanjutkan dengan menghitung total kebutuhan nutrisi sapi berdasarkan variabel berat badan sapi, produksi susu, dan kadar lemak susu. Kemudian membangun populasi awal berdasarkan ukuran populasi, bahan pakan yang dimasukkan, serta batas bobot bahan pakan.



Gambar 4.1 Diagram alir penyelesaian masalah menggunakan IPSO

Berdasarkan Gambar 4.1, setelah membangun populasi awal, proses dilanjutkan dengan perhitungan *fitness* semua partikel. Kemudian disusul dengan proses menentukan posisi lokal terbaik ($pbest_{id}$) dan posisi global terbaik

($gbest_d$). Setelah itu kecepatan partikel akan diperbarui menggunakan persamaan yang telah ditetapkan. Dengan adanya perubahan kecepatan maka posisi partikel juga akan diperbarui. Proses perhitungan *fitness* sampai proses *update* posisi akan diulangi sampai batas iterasi maksimum. Jika iterasi maksimum (T_{max}) telah tercapai maka sistem akan menampilkan keluaran komposisi pakan yang optimal dan program selesai.

4.2.1 Inisialisasi Variabel

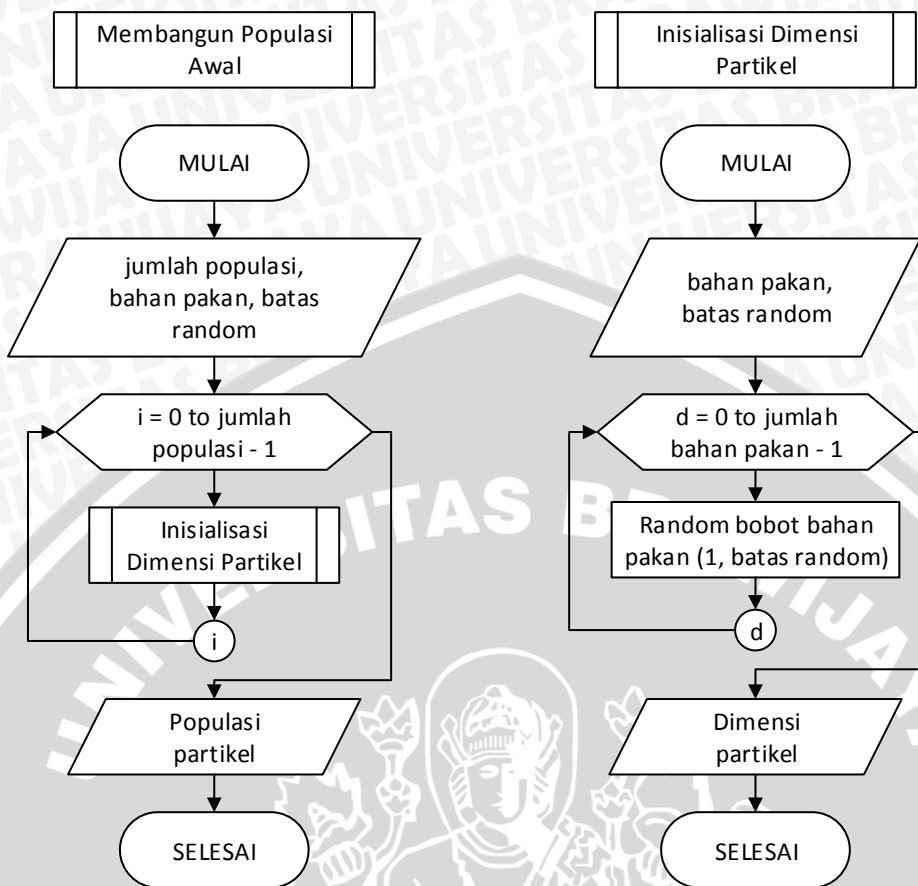
Proses inisialisasi variabel merupakan proses menerima masukan dari *user* berupa variabel pakan, berat badan sapi, produksi susu, kadar lemak susu, dan bahan pakan yang akan diberikan *user* kepada sapi. Ada beberapa variabel yang nilainya secara otomatis disediakan oleh sistem. Variabel tersebut adalah ukuran populasi (p), iterasi maksimum (T_{max}), batas berat bahan pakan, bobot harga, dan bobot *penalty*.

Bobot harga dan bobot *penalty* digunakan untuk menentukan seberapa besar pengaruh harga dan *penalty* dalam menentukan *fitness*. Pada penelitian ini, bobot harga dan bobot *penalty* dibuat sama yaitu 1. Sedangkan untuk nilai ukuran populasi, iterasi maksimum, dan batas berat bahan pakan disesuaikan dengan hasil pengujian. Variabel p , batas bobot maksimum, dan bahan pakan nantinya akan digunakan untuk membangun populasi awal. Sedangkan variabel berat badan sapi, produksi susu, dan kadar lemak digunakan untuk menghitung kebutuhan nutrisi untuk hidup pokok sapi dan kebutuhan nutrisi untuk produksi susu. Dan variabel T_{max} digunakan sebagai batasan siklus dari algoritma IPSO.

4.2.2 Membangun Populasi Awal

Dalam penyelesaian masalah menggunakan algoritma IPSO diperlukan kumpulan partikel yang merepresentasikan solusi dari masalah. Partikel ini dibangkitkan secara acak yang jumlahnya sesuai dengan variabel ukuran populasi. Tiap partikel akan mempunyai beberapa dimensi sesuai jumlah bahan pakan yang dimasukkan. Sebagai contoh *user* memasukkan variabel $p = 5$ dan jumlah bahan pakan yang dipilih ada 3 maka sistem akan membangun partikel sebanyak 5. Tiap partikel tersebut akan mempunyai 3 dimensi sesuai jumlah bahan pakan dimasukkan. Pada tiap dimensi terdapat berat bahan pakan yang nilainya diacak pada interval 1 sampai batas berat maksimum. Nilai berat bahan pakan ini selanjutnya disebut sebagai atribut posisi partikel ke- i pada dimensi ke- d atau dilambangkan sebagai x_{id} .

Pada tiap iterasi t posisi partikel akan diperbarui menurut persamaan 2-5 sehingga lambang posisi partikel menjadi $x_{id}(t)$. Selain atribut posisi, tiap partikel juga mempunyai atribut kecepatan yang dilambangkan sebagai $v_{id}(t)$. Kecepatan partikel juga akan diperbarui pada tiap iterasi menggunakan persamaan 2-8, namun pada tahap inisialisasi kecepatan akan diatur sama dengan 0. Hasil dari proses ini adalah populasi partikel yang akan digunakan untuk proses selanjutnya. Alur dari proses ini juga dapat dijelaskan melalui Gambar 4.2.



Gambar 4.2 Diagram alir proses membangun populasi awal

4.2.3 Menghitung *Fitness*

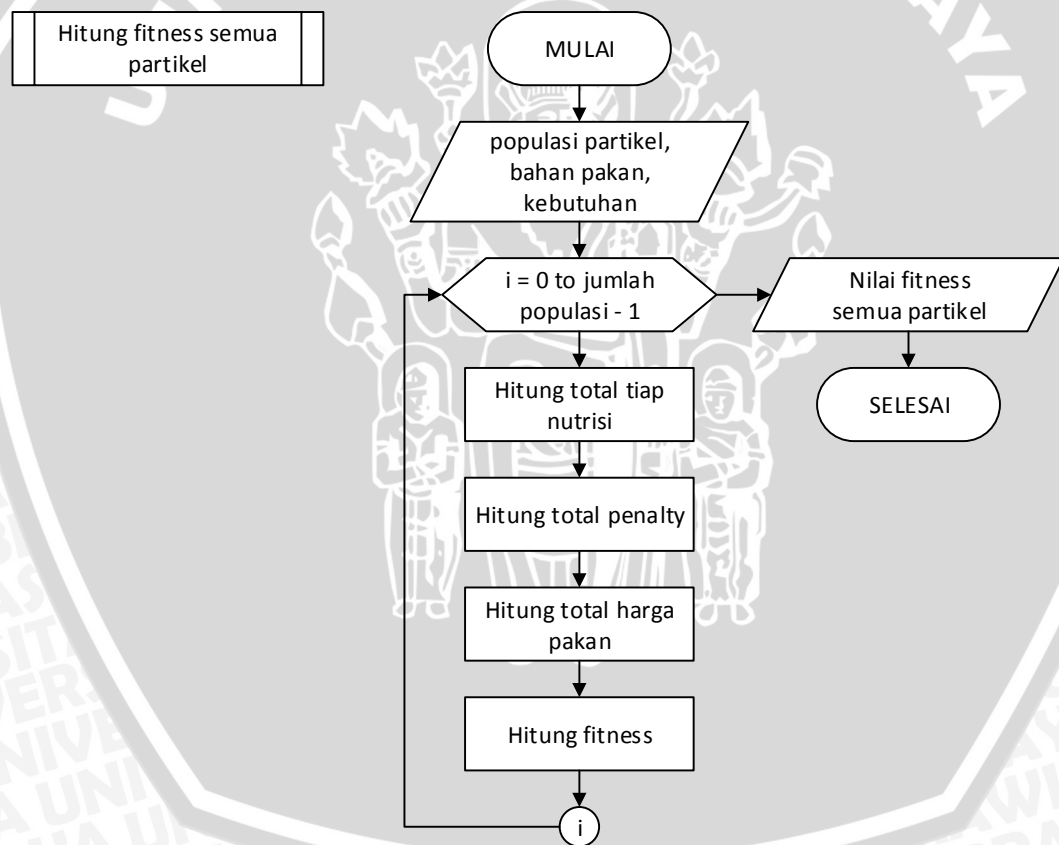
Setelah didapatkan populasi partikel maka selanjutnya adalah perhitungan nilai *fitness*. Nilai *fitness* digunakan untuk mengukur kualitas partikel sebagai representasi dari solusi. Proses ini diawali dengan menghitung nutrisi bahan pakan pada tiap partikel. Perhitungan nutrisi bahan pakan didasarkan pada Tabel 2.5. Kemudian tiap nutrisi dari bahan pakan dikalikan dengan berat bahan pakan (x_{id}). Nutrisi yang sama dari beberapa bahan pakan dalam satu partikel dijumlahkan sehingga membentuk total ketersediaan yang terdiri dari total BK, total PK, total TDN, dan total NE_L . Selain perhitungan nutrisi, total harga pakan juga dihitung karena solusi yang diharapkan adalah pakan yang memenuhi kebutuhan nutrisi sapi dengan total harga yang minimal. Perhitungan harga pakan juga ditentukan berdasarkan Tabel 2.5. Tiap berat bahan pakan (x_{id}) dikalikan dengan harganya kemudian dijumlahkan sehingga dalam satu partikel didapatkan total harga pakan.

Setelah itu menghitung *penalty* untuk mengetahui apakah komposisi bahan pakan pada partikel sudah memenuhi kebutuhan nutrisi sapi atau belum. Nilai *penalty* ditentukan pada tiap nutrisi. Jika nutrisi yang tersedia lebih kecil dari yang dibutuhkan maka *penalty* nutrisi tersebut dihitung dengan cara jumlah kebutuhan dikurangi jumlah ketersediaan. Akan tetapi jika sebaliknya maka nilai *penalty* sama dengan jumlah ketersediaan dikurangi jumlah kebutuhan lalu dikali bobot *penalty*.

Setelah itu menentukan nilai *fitness* yang melibatkan total *penalty* dan total harga pakan menggunakan persamaan 4.1 berikut.

$$fitness = \frac{100000}{w_1 \times total\ harga + w_2 \times (total\ penalty \times 10000)} \quad (4-1)$$

Solusi yang diharapkan dari sistem ini adalah komposisi bahan pakan yang mencukupi kebutuhan sapi dengan total harga pakan yang minimum. Oleh karena itu pada fungsi *fitness* total *penalty* dikalikan dengan 10000. Hal ini supaya total harga dan total *penalty* mempunyai kontribusi yang seimbang dalam menentukan nilai *fitness*. Angka 10000 dipilih karena rata – rata total harga pakan mencapai puluhan ribu. Untuk megimbangi hasil terbagi yang mencapai puluhan ribu maka angka pembagi dipilih 100000 supaya nilai *fitness* yang dihasilkan bisa di atas 0. w_1 pada fungsi *fitness* merupakan bobot untuk total harga, sedangkan w_2 merupakan bobot untuk *penalty*. Pada penelitian ini, kedua bobot tersebut dibuat sama yaitu 1. Pada aplikasi nantinya nilai bobot ini dapat diubah sesuai kehendak *user*. Alur dari proses perhitungan *fitness* dapat dijelaskan melalui Gambar 4.3.

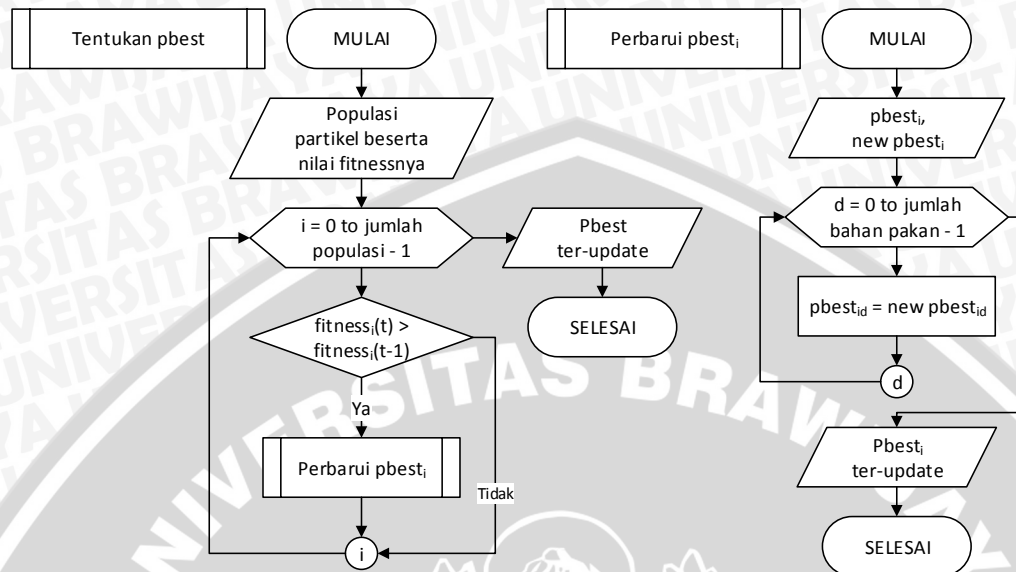


Gambar 4.3 Diagram alir proses menghitung nilai *fitness*

4.2.4 Menentukan Posisi Lokal Terbaik

Tiap partikel pada algoritma PSO memiliki kemampuan untuk mengingat posisi terbaik yang pernah dilaluinya. Posisi lokal terbaik ($pbest_{id}$) diperbarui pada tiap iterasi berdasarkan nilai *fitness* nya. Jika nilai $fitness_i(t)$ lebih besar dari nilai $fitness_pbest_{(i)}(t - 1)$ maka $pbest_{id}$ diubah ke posisi dengan nilai *fitness* yang

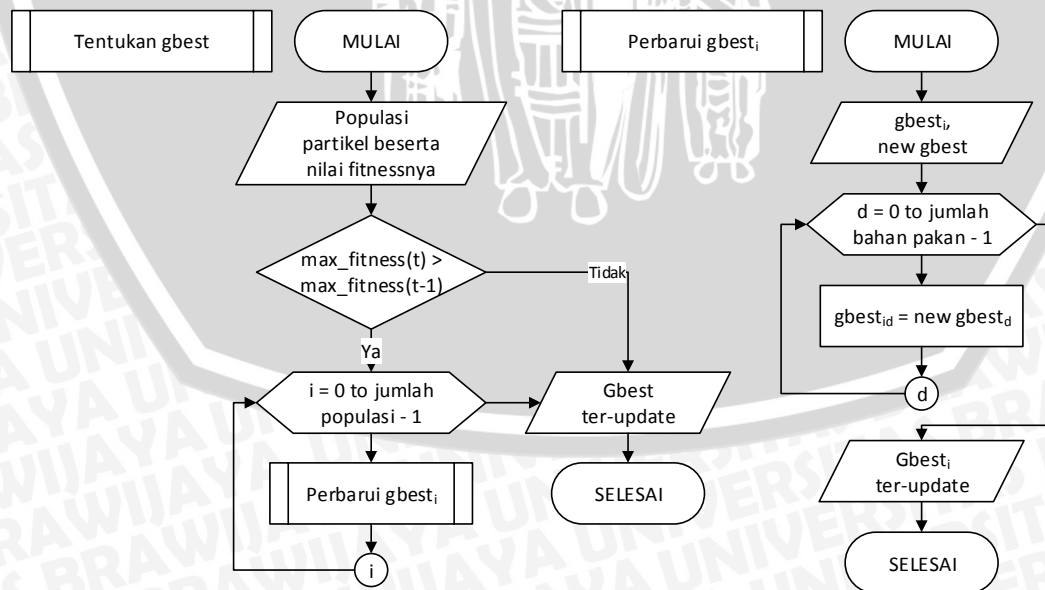
lebih besar. Akan tetapi jika nilai $fitness_i(t)$ lebih kecil sama dengan nilai $fitness_pbest_i(t - 1)$ maka $pbest_{(id)}$ tidak akan mengalami perubahan. Proses ini dapat dijelaskan melalui Gambar 4.4.



Gambar 4.4 Diagram alir proses menentukan posisi lokal terbaik

4.2.5 Menentukan Posisi Global Terbaik

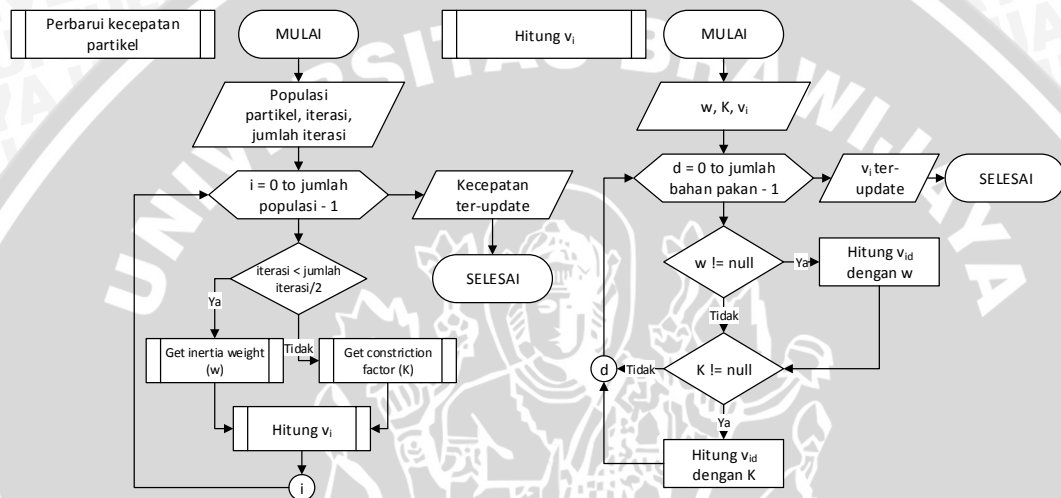
Selain menggunakan penuluran lokal, partikel dalam PSO juga menggunakan penelusuran global dalam upaya mendekati solusi optimum. Penelusuran global dilakukan dengan membandingkan nilai $fitness$ dari seluruh partikel. Posisi partikel dengan nilai $fitness$ paling tinggi diantara semua partikel akan menjadi posisi global terbaik ($gbest_d$). Dengan kata lain $gbest_d$ sama dengan $pbest_{id}$ terbaik dari seluruh partikel. Diagram alir dari proses ini dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram alir menentukan posisi global terbaik

4.2.6 Memperbarui Kecepatan Partikel

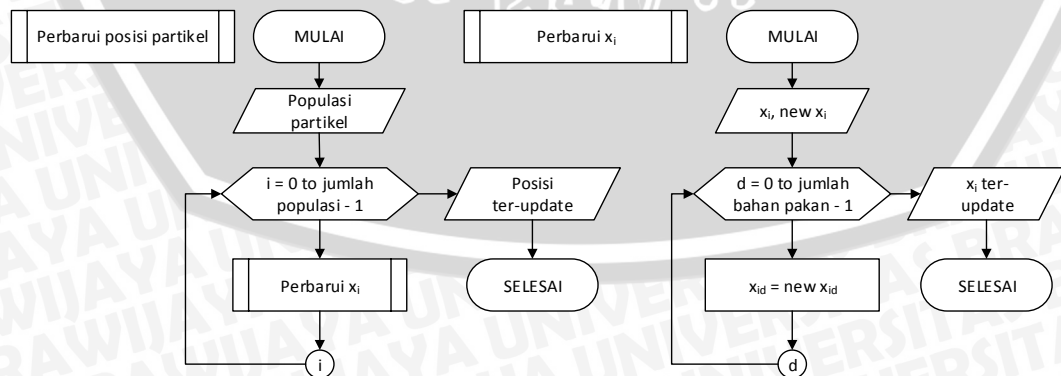
Seperti yang dijelaskan pada sub-bab 2.3.1 bahwa partikel bergerak dengan kecepatan yang dinamis untuk mendekati solusi optimum. Dalam menentukan kecepatannya, partikel menggunakan posisi lokal terbaik ($pbest_{id}$) dan posisi global terbaik ($gbest_d$) yang telah didapatkan dari proses sebelumnya. Proses menentukan kecepatan pada penelitian ini menggunakan persamaan kecepatan yang telah di-improve, yaitu persamaan 2-8. Berdasarkan persamaan tersebut, jika iterasi kurang dari jumlah iterasi dibagi 2 maka akan digunakan komponen *inertia weight* (w). Sedangkan jika iterasi lebih dari sama dengan jumlah iterasi dibagi 2 maka digunakan komponen *constriction factor* (K). Alur dari proses ini dapat dijelaskan melalui Gambar 4.6.



Gambar 4.6 Diagram alir menentukan kecepatan partikel

4.2.7 Memperbarui Posisi Partikel

Setelah kecepatan partikel diperbarui maka selanjutnya memperbarui posisi partikel menggunakan persamaan 2-5. Posisi partikel yang telah diperbarui akan digunakan sebagai populasi pada iterasi selanjutnya. Alur memperbarui posisi partikel dapat dijelaskan melalui Gambar 4.7.



Gambar 4.7 Diagram alir menentukan posisi partikel

4.3 Contoh Penyelesaian Masalah Menggunakan Algoritma IPSO

Pada sub bab ini akan dijelaskan manualisasi penyelesaian kasus optimasi komposisi pakan sapi perah menggunakan algoritma IPSO. Proses penyelesaian akan berjalan sesuai dengan alur penyelesaian masalah yang telah dijelaskan pada sub – bab 4.2.

4.3.1 Inisialisasi Variabel

Berdasarkan permasalahan yang telah didefinisikan pada sub-bab 4.1 maka inisialisasi variabel adalah sebagai berikut,

- Berat badan sapi : 450 Kg
- Jumlah produksi susu : 15 L
- Kadar lemak susu : 4,5 %
- Ukuran populasi (p) : 5
- Jumlah iterasi (T_{max}) : 2
- Batas berat maksimum : 20 Kg
- Bahan Pakan :
 - ✓ Rumput gajah
 - ✓ Tebon jagung
 - ✓ Bapro

4.3.2 Menghitung Kebutuhan Nutrisi Sapi Perah

Kebutuhan nutrisi sapi perah meliputi kebutuhan untuk hidup pokok, produksi, dan pertumbuhan. Sesuai dengan Tabel 2.2, kebutuhan hidup pokok sapi dengan berat badan 450 kg memerlukan NE_L sebesar 7,82 Mkal, TDN sebesar 3,44 kg, dan PK sebesar 403 g. Kemudian berdasarkan Tabel 2.3, untuk memproduksi 1 L susu dengan kadar lemak sebesar 4,5 % dibutuhkan NE_L sebesar 0,78 Mkal, TDN sebesar 0,344 kg, dan PK sebesar 92 g. Jumlah tersebut untuk memenuhi produksi 1 L susu, maka untuk produksi susu sebanyak 15 L dibutuhkan

- NE_L = jumlah produksi susu x kebutuhan NE_L untuk produksi 1 L susu
 = 15 L x 0,78 Mkal
 = 11,7 Mkal
- TDN = jumlah produksi susu x kebutuhan TDN untuk produksi 1 L susu
 = 15 L x 0,344 Kg
 = 5,16 Kg
- PK = jumlah produksi susu x kebutuhan PK untuk produksi 1 L susu
 = 15 L x 0,092 Kg
 = 1,38 Kg

Selanjutnya menghitung kebutuhan pertumbuhan sapi berdasarkan persamaan 2-1. Telah diketahui bahwa kebutuhan NE_L untuk hidup pokok sapi sebesar 7,82 Mkal, maka

$$\begin{aligned} \text{Pertumbuhan} &= 10\% \times \text{NE}_L \text{ hidup pokok} \\ &= 10\% \times 7,82 \text{ Mkal} \\ &= 0,782 \text{ Mkal} \end{aligned}$$

Kebutuhan NE_L untuk hidup pokok sapi sebesar 7,82 Mkal, sedangkan kebutuhan NE_L untuk memproduksi 15 L susu sebesar 11,7 Mkal, dan kebutuhan NE_L untuk pertumbuhan sebesar 0,782 Mkal. Jika ketiga kebutuhan NE_L ini dijumlahkan maka

$$\begin{aligned} \text{Total } \text{NE}_L &= \text{NE}_L \text{ hidup pokok} + \text{NE}_L \text{ produksi} + \text{NE}_L \text{ pertumbuhan} \\ &= 7,82 \text{ Mkal} + 11,7 \text{ Mkal} + 0,782 \text{ Mkal} \\ &= 20,302 \text{ Mkal} \end{aligned}$$

Setelah mendapatkan total NE_L maka langkah selanjutnya adalah menghitung kebutuhan BK. Berdasarkan Tabel 2.4, produksi susu sebesar 15 L memerlukan 1,62 NE_L/kg BK, sehingga

$$\begin{aligned} \text{BK} &= \text{total } \text{NE}_L / \text{kebutuhan } \text{NE}_L \text{ per kg BK} \\ &= 20,302 \text{ Mkal} / 1,62 \text{ Mkal per kg BK} \\ &= 12,532098765432 \text{ kg} \end{aligned}$$

Dari perhitungan yang telah dilakukan, didapatkan total kebutuhan nutrisi sapi seperti pada Tabel 4.1.

Tabel 4.1 Total kebutuhan nutrisi sapi

Kebutuhan	NE_L (Mkal)	TDN (kg)	PK (kg)	BK (kg)
Hidup pokok	7,82	3,44	0,403	12,532098765432
Produksi	11,7	5,16	1,38	
Pertumbuhan	0,782	-	-	
Total	20,302	8,6	1,783	12,532098765432

4.3.3 Membangun Populasi Awal

Langkah selanjutnya yaitu membangun populasi partikel sesuai ukuran populasi dan jumlah jenis bahan pakan. Ukuran populasi yang digunakan adalah 5 maka jumlah partikel yang akan dibangkitkan sebanyak 5. Kemudian tiap partikel mempunyai 3 dimensi karena jenis bahan pakan yang dimasukkan ada 3. Tiap dimensi mempunyai berat bahan pakan yang nilainya diacak pada rentang 1 sampai batas berat maksimum (20). Selain berat bahan pakan, tiap dimensi juga memiliki atribut kecepatan. Di awal iterasi, kecepatan tiap dimensi diatur sama dengan 0. Tabel 4.2 merupakan populasi awal partikel yang akan digunakan untuk proses selanjutnya.

Tabel 4.2 Populasi awal partikel

Partikel	Rumput Gajah (Kg)	Tebon Jagung (Kg)	Bapro (Kg)
	x_{i1}	x_{i2}	x_{i3}
1	8,34	7,30	2,21
2	5,92	3,43	11,20
3	4,34	4,61	11,41
4	2,62	5,86	11,64
5	9,1	1,14	10,65

4.3.4 Menghitung *Fitness*

Setelah didapatkan populasi awal maka selanjutnya tiap partikel dihitung nilai *fitness* nya. Perhitungan nilai *fitness* digunakan untuk mengetahui apakah komposisi bahan pakan pada tiap partikel sudah mencukupi kebutuhan nutrisi sapi atau belum. Berdasarkan Tabel 2.5 dan persamaan 2-2, berikut ini merupakan perhitungan nutrisi dan harga bahan pakan pada partikel ke-1.

- Rumput gajah
 - Berat = 8,34 kg
 - BK = berat rumput gajah x kandungan BK rumput gajah
= 8,34 kg x 0,365
= 3,0441 kg
 - PK = berat rumput gajah x kandungan PK rumput gajah
= 8,34 kg x 0,128
= 1,06752 kg
 - TDN = berat rumput gajah x kandungan TDN rumput gajah
= 8,34 kg x 0,557
= 4,64538 kg
 - NE_L = (0,0245 x TDN rumput gajah (%) – 0,12) x berat rumput gajah
= (0,0245 x 55,7 – 0,12) x 8,34
= 10,380381 Mkal
 - Harga = berat rumput gajah x harga rumput gajah
= 8,34 kg x Rp 1250
= Rp 10425
- Tebon jagung
 - Berat = 7,30 kg

$$\begin{aligned} \text{BK} &= \text{berat tebon jagung} \times \text{kandungan BK tebon jagung} \\ &= 7,30 \text{ kg} \times 0,351 \\ &= 2,5623 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{PK} &= \text{berat tebon jagung} \times \text{kandungan PK tebon jagung} \\ &= 7,30 \text{ kg} \times 0,088 \\ &= 0,6424 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{TDN} &= \text{berat tebon jagung} \times \text{kandungan TDN tebon jagung} \\ &= 7,30 \text{ kg} \times 0,688 \\ &= 5,0224 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{NE}_L &= (0,0245 \times \text{TDN tebon jagung} (\%) - 0,12) \times \text{berat tebon jagung} \\ &= (0,0245 \times 68,8 - 0,12) \times 7,30 \\ &= 11,42888 \text{ Mkal} \end{aligned}$$

$$\begin{aligned} \text{Harga} &= \text{berat tebon jagung} \times \text{harga tebon jagung} \\ &= 7,30 \text{ kg} \times \text{Rp } 1700 \\ &= \text{Rp } 12410 \end{aligned}$$

- Bapro

$$\text{Berat} = 2,21 \text{ kg}$$

$$\begin{aligned} \text{BK} &= \text{berat bapro} \times \text{kandungan BK bapro} \\ &= 2,21 \text{ kg} \times 0,8179 \\ &= 1,807559 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{PK} &= \text{berat bapro} \times \text{kandungan PK bapro} \\ &= 2,21 \text{ kg} \times 0,1869 \\ &= 0,413049 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{TDN} &= \text{berat bapro} \times \text{kandungan TDN bapro} \\ &= 2,21 \text{ kg} \times 0,32456 \\ &= 0,7172776 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{NE}_L &= (0,0245 \times \text{TDN bapro} (\%) - 0,12) \times \text{berat bapro} \\ &= (0,0245 \times 32,456 - 0,12) \times 2,21 \\ &= 1,49213012 \text{ Mkal} \end{aligned}$$

$$\begin{aligned} \text{Harga} &= \text{berat bapro} \times \text{harga bapro} \\ &= 2,21 \text{ kg} \times \text{Rp } 3000 \\ &= \text{Rp } 6630 \end{aligned}$$

Setelah didapatkan jumlah nutrisi pada masing-masing bahan pakan, selanjutnya pada nutrisi yang sama dari ketiga bahan pakan dijumlahkan seperti berikut,

- Total BK = jumlah BK rumput gajah + jumlah BK tebon jagung + jumlah BK bapros
 = 3,0441 kg + 2,5623 kg + 1,807559 kg
 = 7,413959 kg
- Total PK = jumlah PK rumput gajah + jumlah PK tebon jagung + jumlah PK bapros
 = 1,06752 kg + 0,6424 kg + 0,413049 kg
 = 2.122969 kg
- Total TDN = jumlah TDN rumput gajah + jumlah TDN tebon jagung + jumlah TDN bapros
 = 4,64538 kg + 5,0224 kg + 0,7172776 kg
 = 10,3850576 kg
- Total NE_L = jumlah NE_L rumput gajah + jumlah NE_L tebon jagung + jumlah NE_L bapros
 = 10,380381 Mkal + 11,42888 Mkal + 1,49213012 Mkal
 = 23,30139112 Mkal

Total masing – masing nutrisi yang telah didapatkan disebut sebagai total ketersediaan nutrisi partikel ke-1 sebagaimana yang ditampilkan pada tabel 4.3.

Tabel 4.3 Total ketersediaan nutrisi partikel ke-1

Partikel	Total Nutrisi			
	BK (Kg)	PK (Kg)	TDN (Kg)	NE _L (Mkal)
1	7,413959	2.122969	10,3850576	23,30139112

Setelah perhitungan nutrisi bahan pakan, langkah selanjutnya ialah menentukan *penalty* dengan cara menghitung selisih antara total ketersediaan dengan total kebutuhan. Total ketersediaan nutrisi pada partikel ke-1 dijelaskan melalui Tabel 4.3. Sedangkan total kebutuhan nutrisi sapi dijelaskan pada Tabel 4.1. Pada nutrisi yang sama, nilai nutrisi pada Tabel 4.3 dibandingkan dengan nutrisi yang ada pada Tabel 4.1 seperti berikut,

- *Penalty* BK

Berdasarkan Tabel 4.1 dan Tabel 4.3 diketahui bahwa jumlah BK yang dibutuhkan lebih besar dari yang tersedia maka

$$\begin{aligned} \text{Penalty BK} &= \text{jumlah kebutuhan BK} - \text{jumlah ketersediaan BK} \\ &= 12,532098765432 \text{ kg} - 7,413959 \text{ kg} \\ &= 5,118139765 \end{aligned}$$

- **Penalty PK**

Berdasarkan Tabel 4.1 dan Tabel 4.3 diketahui bahwa jumlah PK yang dibutuhkan lebih kecil dari yang tersedia maka nilai *penalty* dikalikan dengan 0,9 seperti berikut.

$$\begin{aligned} \text{Penalty PK} &= (\text{jumlah ketersediaan PK} - \text{jumlah kebutuhan PK}) \times \text{bobot } \textit{penalty} \\ &= (2.122969 \text{ kg} - 1,783 \text{ kg}) \times 0,9 \\ &= 0,3059721 \end{aligned}$$

- **Penalty TDN**

Berdasarkan Tabel 4.1 dan Tabel 4.3 diketahui bahwa jumlah TDN yang dibutuhkan lebih kecil dari yang tersedia maka nilai *penalty* dikalikan dengan 0,9 seperti berikut.

$$\begin{aligned} \text{Penalty TDN} &= (\text{jumlah ketersediaan TDN} - \text{jumlah kebutuhan TDN}) \times \\ &\quad \text{bobot } \textit{penalty} \\ &= (10,3850576 \text{ kg} - 8,6 \text{ kg}) \times 0,9 \\ &= 1,60655184 \end{aligned}$$

- **Penalty NE_L**

Berdasarkan Tabel 4.1 dan Tabel 4.3 diketahui bahwa jumlah NE_L yang dibutuhkan lebih kecil dari yang tersedia maka nilai *penalty* dikalikan dengan 0,9 seperti berikut.

$$\begin{aligned} \text{Penalty NE}_L &= (\text{jumlah ketersediaan NE}_L - \text{jumlah kebutuhan NE}_L) \times \\ &\quad \text{bobot } \textit{penalty} \\ &= (23,30139112 \text{ Mkal} - 20,302 \text{ Mkal}) \times 0,9 \\ &= 2,699452008 \end{aligned}$$

Kemudian keempat *penalty* tersebut dijumlahkan menjadi total *penalty* partikel ke-1 seperti berikut,

$$\begin{aligned} \text{Total } \textit{penalty} &= \textit{penalty BK} + \textit{penalty PK} + \textit{penalty TDN} + \textit{penalty NE}_L \\ &= 5,118139765 + 0,3059721 + 1,60655184 + 2,699452008 \\ &= 9,730115713 \end{aligned}$$

Langkah selanjutnya yaitu menghitung total harga pakan pada partikel ke-1.

$$\begin{aligned}
 \text{Total harga} &= \text{total harga rumput gajah} + \text{total harga tebon jagung} + \text{total} \\
 &\quad \text{harga bapro} \\
 &= \text{Rp } 10425 + \text{Rp } 12410 + \text{Rp } 6630 \\
 &= \text{Rp } 29465
 \end{aligned}$$

Setelah didapatkan total *penalty* dan total harga maka dapat ditentukan nilai *fitness* partikel ke-1 menggunakan persamaan 4-1.

$$\begin{aligned}
 fitness_1 &= \frac{100000}{w_1 \times \text{total harga} + w_2 \times (\text{total penalty} \times 10000)} \\
 &= \frac{100000}{1 \times 29465 + 1 \times (9,730115713 \times 10000)} \\
 &= 0,788854
 \end{aligned}$$

Perhitungan nilai *fitness* dilakukan pada semua partikel sehingga didapatkan nilai *fitness* pada masing-masing partikel seperti pada Tabel 4.4.

Tabel 4.4 Nilai *fitness* pada tiap partikel

Partikel	Rumput Gajah (Kg)	Tebon Jagung (Kg)	Bapro (Kg)	Nilai <i>fitness</i>
	x_{i1}	x_{i2}	x_{i3}	
1	8,34	7,30	2,21	0,78885
2	5,92	3,43	11,20	1,52725
3	4,34	4,61	11,41	1,52141
4	2,62	5,86	11,64	1,52770
5	9,1	1,14	10,65	1,51736

4.3.5 Menentukan Posisi Lokal Terbaik

Pada awal iterasi, posisi partikel pertama akan menjadi posisi lokal terbaik ($pbest_{id}$) dan nilai *fitness* nya akan menjadi nilai $fitness_pbest_i$. Akan tetapi untuk iterasi selanjutnya, tiap partikel akan membandingkan nilai *fitness* nya yang terbaru dengan nilai $fitness_pbest$ nya yang sebelumnya. Hanya posisi partikel dengan nilai *fitness* yang lebih baik yang akan menjadi $pbest_{id}$. Jika nilai *fitness* lebih kecil atau tetap maka $pbest_{id}$ dan $fitness_pbest_i$ tidak akan diperbarui. Tabel 4.5 berikut merupakan posisi lokal terbaik pada tiap partikel untuk iterasi pertama.

Tabel 4.5 Posisi lokal terbaik pada tiap partikel

Partikel	Rumput Gajah (Kg)	Tebon Jagung (Kg)	Bapro (Kg)	Nilai $fitness_pbest_i$
	$pbest_{i1}$	$pbest_{i2}$	$pbest_{i3}$	
1	8,34	7,30	2,21	0,78885
2	5,92	3,43	11,20	1,52725
3	4,34	4,61	11,41	1,52141
4	2,62	5,86	11,64	1,52770
5	9,1	1,14	10,65	1,51736

4.3.6 Menentukan Posisi Global Terbaik

Posisi global terbaik ($gbest_a$) ditentukan berdasarkan nilai $fitness_pbest$ terbaik dari seluruh partikel. Dari Tabel 4.5 diketahui bahwa nilai $fitness_pbest$ tertinggi dimiliki oleh partikel 4 maka setiap partikel akan menyimpan posisi partikel ke-4 sebagai posisi global terbaik. Begitu juga dengan nilai $fitness_pbest$ dari partikel ke-4 akan menjadi nilai $fitness_gbest$ pada tiap partikel. Tabel 4.8 berikut menampilkan posisi global terbaik pada tiap partikel untuk iterasi pertama.

Tabel 4.6 Posisi global terbaik pada tiap partikel

Partikel	Rumput Gajah (Kg)	Tebon Jagung (Kg)	Bapro (Kg)	Nilai $fitness_gbest_i$
	$gbest_{i1}$	$gbest_{i2}$	$gbest_{i3}$	
1	2,62	5,86	11,64	1,52770
2	2,62	5,86	11,64	1,52770
3	2,62	5,86	11,64	1,52770
4	2,62	5,86	11,64	1,52770
5	2,62	5,86	11,64	1,52770

Posisi global terbaik akan diperbarui jika nilai $fitness_pbest$ tertinggi pada iterasi selanjutnya lebih besar dari $fitness_gbest$ saat ini.

4.3.7 Memperbarui Kecepatan Partikel

Proses selanjutnya yaitu memperbarui kecepatan tiap dimensi pada tiap partikel menggunakan persamaan 2-8. Seperti yang dijelaskan pada bab 4.2.3, kecepatan partikel awal mulanya diatur sama dengan 0. Kemudian pada tiap iterasi akan diperbarui menggunakan persamaan 2-8. Berdasarkan persamaan tersebut, ketika iterasi kurang dari jumlah iterasi dibagi 2 maka digunakan komponen *inertia weight* (w). Dalam menentukan komponen ini, partikel akan diseleksi apakah posisi partikel tersebut sama dengan posisi global terbaik. Untuk partikel ke-1, posisinya bukan merupakan posisi global terbaik maka sesuai dengan persamaan 2-6,

$$\begin{aligned}
 w &= 0,857143 + (1 - 0,857143) \times \left(1 - \frac{t}{T_{max}}\right) \\
 &= 0,857143 + (1 - 0,857143) \times \left(1 - \frac{0}{2}\right) \\
 &= 1
 \end{aligned}$$

Setelah didapatkan *inertia weight* selanjutnya dapat ditentukan kecepatan tiap dimensi pada partikel ke-1 seperti berikut,

$$\begin{aligned}
 v_{11} &= w \times v_{11} + 2 \times r_1 \times (pbest_{11} - x_{11}) + 2 \times r_2 \times (gbest_1 - x_{11}) \\
 &= 1 \times 0 + 2 \times r_1 \times (8,34 - 8,34) - 2 \times r_2 \times (2,62 - 8,34) \\
 &= -7,48 \\
 v_{12} &= w \times v_{12} + 2 \times r_1 \times (pbest_{12} - x_{12}) + 2 \times r_2 \times (gbest_2 - x_{12}) \\
 &= 1 \times 0 + 2 \times r_1 \times (7,30 - 7,30) - 2 \times r_2 \times (5,86 - 7,30) \\
 &= -1,88
 \end{aligned}$$

$$\begin{aligned}
 v_{13} &= w \times v_{13} + 2 \times r_1 \times (pbest_{13} - x_{13}) + 2 \times r_2 \times (gbest_3 - x_{13}) \\
 &= 1 \times 0 + 2 \times r_1 \times (2,21 - 2,21) - 2 \times r_2 \times (11,64 - 2,21) \\
 &= 12,34
 \end{aligned}$$

r_1 maupun r_2 merupakan angka *random* antara 0 sampai 1 yang diacak pada tiap iterasi. Hasil perhitungan kecepatan untuk semua partikel ditampilkan pada Tabel 4.7.

Tabel 4.7 Kecepatan partikel terbaru

Partikel	Rumput Gajah	Tebon Jagung	Bapro
	v_{i1}	v_{i2}	v_{i3}
1	-7,48	-1,88	12,34
2	-4,32	3,18	0,58
3	-2,25	1,64	0,30
4	0,00	0,00	0,00
5	-8,48	6,18	1,30

Tiap partikel akan menyimpan kecepatan terbarunya untuk digunakan pada proses selanjutnya.

4.3.8 Memperbarui Posisi Partikel

Setelah kecepatan partikel diperbarui maka selanjutnya posisi partikel diperbarui menggunakan persamaan 2-5. Perubahan posisi pada partikel ke-1 dapat dilihat pada perhitungan berikut,

$$\begin{aligned}
 x_{11}(t+1) &= x_{11} + v_{11} = 8,34 + (-7,48) = 0,86 \\
 x_{12}(t+1) &= x_{12} + v_{12} = 7,30 + (-1,88) = 5,42 \\
 x_{13}(t+1) &= x_{13} + v_{13} = 2,21 + 12,34 = 14,55
 \end{aligned}$$

Untuk perubahan posisi pada tiap partikel dapat dilihat pada Tabel 4.8 berikut.

Tabel 4.8 Posisi partikel terbaru

Partikel	Rumput Gajah (Kg)	Tebon Jagung (Kg)	Bapro (Kg)
	$x_{i1}(t+1)$	$x_{i2}(t+1)$	$x_{i3}(t+1)$
1	0,86	5,42	14,55
2	1,60	6,61	11,78
3	2,09	6,25	11,71
4	2,62	5,86	11,64
5	0,62	7,32	11,95

Himpunan partikel pada Tabel 4.8 akan menjadi populasi partikel pada iterasi selanjutnya.

4.4 Perancangan Pengujian

Pengujian ini bertujuan untuk menentukan parameter IPSO yang tepat sehingga dapat dihasilkan solusi yang mendekati optimum. Adapun parameter algoritma IPSO yang diuji antara lain jumlah iterasi, ukuran populasi, dan interval

berat bahan pakan. Seluruh pengujian parameter ini menggunakan variabel objek yang sama seperti berikut.

- Berat badan sapi : 450 Kg
- Jumlah produksi susu : 15 L
- Kadar lemak susu : 4,5 %
- Bahan Pakan :
 - ✓ Rumput gajah
 - ✓ Tebon jagung
 - ✓ Bapro

4.4.1 Perancangan Pengujian Jumlah Iterasi

Pengujian jumlah iterasi digunakan untuk mendapatkan parameter jumlah iterasi yang tepat guna mendapatkan komposisi pakan sapi perah yang optimal. Data uji untuk jumlah iterasi dimulai dari 20 sampai 160 dengan kelipatan 20. Algoritma berbasis PSO merupakan algoritma yang bersifat stokastik dimana pada setiap dijalankan akan menghasilkan keluaran yang berbeda (Mahmudy, 2015). Untuk itu pengujian akan dilakukan sebanyak 5 kali pada setiap data uji guna mendapatkan hasil yang valid. Maka proses pengujian jumlah iterasi dilakukan berdasarkan rancangan pengujian pada Tabel 4.9.

Tabel 4.9 Rancangan pengujian jumlah iterasi

Jumlah iterasi	Nilai <i>fitness</i> pada percobaan					Rata - rata
	1	2	3	4	5	
20						
40						
60						
80						
100						
120						
140						
160						

4.4.2 Perancangan Pengujian Ukuran Populasi

Pengujian ukuran populasi digunakan untuk mendapatkan ukuran populasi yang tepat guna mendapatkan komposisi pakan sapi perah yang optimal. Data uji untuk ukuran populasi dimulai dari 20 sampai 180 dengan kelipatan 20. Proses pengujian juga dilakukan sebanyak 5 kali pada setiap data uji guna mendapatkan hasil yang valid. Proses pengujian ukuran populasi dilakukan berdasarkan rancangan pengujian pada Tabel 4.10.

Tabel 4.10 Rancangan pengujian ukuran populasi

Ukuran populasi	Nilai <i>fitness</i> pada percobaan					Rata - rata
	1	2	3	4	5	
20						
40						

Ukuran populasi	Nilai <i>fitness</i> pada percobaan					Rata - rata
	1	2	3	4	5	
60						
80						
100						
120						
140						
160						
180						

4.4.3 Perancangan Pengujian Interval Berat Bahan Pakan

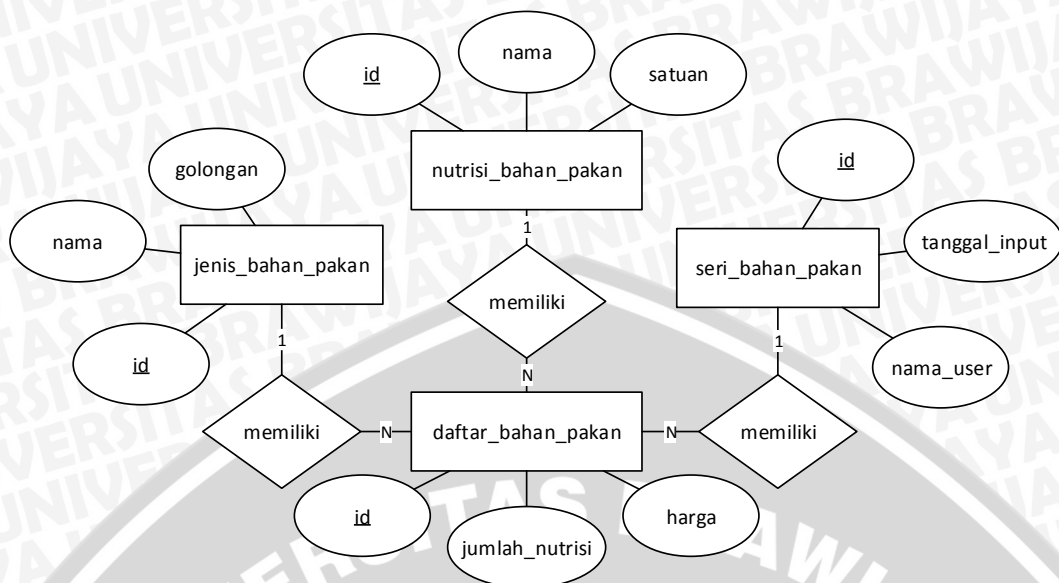
Pengujian ini digunakan untuk mendapatkan parameter interval berat bahan pakan yang tepat guna mendapatkan komposisi pakan sapi perah yang optimal. Interval berat bahan pakan berfungsi sebagai batasan untuk mengacak berat bahan pakan pada saat pembuatan populasi awal. Data uji untuk parameter ini dimulai dari interval 1 - 10 sampai interval 1 - 25 dengan kelipatan berat maksimum yaitu 5. Pengujian juga dilakukan sebanyak 5 kali pada setiap data uji guna mendapatkan hasil yang valid. Proses pengujian dilakukan berdasarkan rancangan pengujian pada Tabel 4.11.

Tabel 4.11 Rancangan pengujian interval berat bahan pakan

Interval	Nilai <i>fitness</i> pada percobaan					Rata - rata
	1	2	3	4	5	
1-10						
1-15						
1-20						
1-25						

4.5 Perancangan Basis Data

Pada penelitian ini, basis data digunakan untuk menyimpan data bahan pakan. Data yang disimpan meliputi jenis bahan pakan, jumlah nutrisi yang terkandung dalam bahan pakan dan harga bahan pakan. Sistem akan mengakses basis data ketika proses optimasi untuk mendapatkan data nutrisi dari bahan pakan yang dipilih *user*. Dengan adanya basis data ini, sistem juga dapat menyimpan data bahan pakan yang dimasukkan oleh *user*. Selain itu *user* juga dapat mengubah jumlah nutrisi dan harga bahan pakan pada data yang sudah ada. Gambar 4.8 merupakan rancangan basis data yang akan diimplementasikan. Ada empat entitas pada rancangan basis data antara lain *nutrisi_bahan_pakan*, *jenis_bahan_pakan*, *seri_bahan_pakan*, dan *daftar_bahan_pakan*. Tiap entitas memiliki atribut. Nama atribut yang digaris bawahi menandakan bahwa atribut tersebut merupakan *key* dari entitas. Bentuk belah ketupat pada rancangan basis data menyatakan relasi antar entitas.



Gambar 4.8 Entity Relationship Diagram (ERD)

4.6 Perancangan Antarmuka

Perancangan antarmuka menjelaskan rancangan antarmuka yang akan diimplementasikan. Terdapat tiga rancangan antarmuka yang akan diimplementasikan, antara lain rancangan antarmuka *input* data, rancangan antarmuka untuk pengelolaan data bahan pakan, dan rancangan antarmuka untuk menambah dan mengubah data bahan pakan.

4.6.1 Perancangan Antarmuka *Input* Data

Antarmuka *input* data digunakan untuk memasukkan variabel yang dibutuhkan untuk proses optimasi. Pada halaman ini juga akan ditampilkan kebutuhan nutrisi sapi serta hasil perhitungan pada tiap iterasi. Hasil dari optimasi juga akan ditampilkan pada halaman ini.

1 SISTEM OPTIMASI KOMPOSISI PAKAN SAPI PERAH MENGGUNAKAN IMPROVED PARTICLE SWARM OPTIMIZATION (IPSO)

Optimasi Bahan Pakan Data Bahan Pakan <div style="border: 1px solid gray; height: 100px; margin-top: 10px; display: flex; align-items: center; justify-content: center;">2</div>	Enter Text 3 Enter Text 4	<input checked="" type="checkbox"/> Hijauan A <input checked="" type="checkbox"/> Hijauan B <input checked="" type="checkbox"/> Hijauan C 5	<input checked="" type="checkbox"/> Konsentrat A <input checked="" type="checkbox"/> Konsentrat B <input checked="" type="checkbox"/> Konsentrat C 6
<input style="width: 100%;" type="text"/> 7			
<div style="border: 1px solid gray; height: 30px; display: flex; align-items: center; justify-content: center;">8</div>		<div style="border: 1px solid gray; height: 30px; display: flex; align-items: center; justify-content: center;">9</div>	
<div style="border: 1px solid gray; height: 30px; display: flex; align-items: center; justify-content: center;">10</div>			
<div style="border: 1px solid gray; height: 30px; display: flex; align-items: center; justify-content: center;">11</div>			

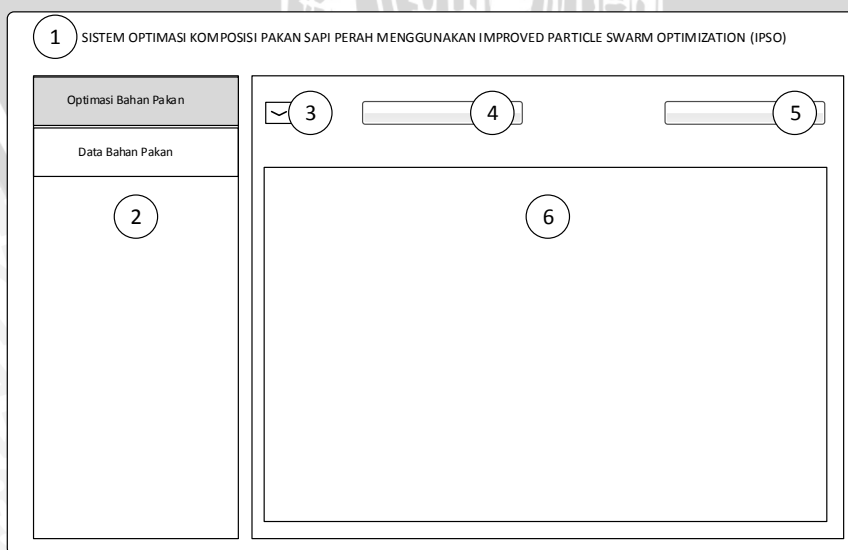
Gambar 4.9 Rancangan antarmuka *input* data

Keterangan rancangan antarmuka *input* data pada Gambar 4.9 adalah sebagai berikut,

1. *Header* aplikasi.
2. *Sidebar* yang menampilkan dua menu utama aplikasi yaitu menu optimasi bahan pakan dan menu data bahan pakan. Menu dengan warna putih artinya menu tersebut sedang aktif.
3. *Text box* untuk memasukkan variabel berat badan sapi, jumlah produksi susu, dan kadar lemak susu.
4. *Text box* untuk memasukkan parameter IPSO yang terdiri dari ukuran populasi, jumlah iterasi, dan batas berat tiap bahan pakan. Dan juga bobot harga dan bobot *penalty*.
5. *Checkbox* untuk memilih bahan pakan hijauan.
6. *Checkbox* untuk memilih bahan pakan konsentrat.
7. *Button* untuk memulai proses optimasi.
8. Tabel untuk menampilkan kebutuhan nutrisi sapi berdasarkan parameter yang dimasukkan.
9. Tabel yang menampilkan bahan pakan yang dipilih *user* beserta kandungan nutrisi dan harganya.
10. Grafik yang menampilkan *pbest* dan *gbest* pada tiap iterasi.
11. Tabel yang menampilkan hasil optimum dari proses optimasi.

4.6.2 Perancangan Antarmuka Pengelolaan Data Bahan Pakan

Antarmuka pengelolaan data bahan pakan merupakan halaman untuk melihat, menambah, dan mengubah data bahan pakan. Gambar 4.10 berikut merupakan rancangan antarmuka untuk pengelolaan data bahan pakan.



Gambar 4.10 Rancangan antarmuka pengelolaan data bahan pakan

Adapun keterangan dari rancangan antarmuka pada Gambar 4.10 adalah sebagai berikut,

1. *Header* aplikasi.
2. *Sidebar* yang menampilkan menu optimasi bahan pakan dan menu data bahan pakan. Menu data bahan berwarna putih yang artinya menu tersebut sedang aktif.
3. *Drop down* untuk memilih seri bahan pakan.
4. *Button* untuk menampilkan data bahan pakan berdasarkan seri yang dipilih *user*.
5. *Button* untuk menambah data bahan pakan baru.
6. Tabel yang menampilkan data bahan pakan. Pada tiap baris data bahan pakan, terdapat *button* untuk mengubah data bahan pakan.

4.6.3 Perancangan Antarmuka *Input* dan *Edit* Data Bahan Pakan

Antarmuka *input* dan *edit* data bahan pakan digunakan untuk menambah dan mengubah data bahan pakan. Antarmuka ini akan muncul setelah *user* menekan *button* nomor 5 pada Gambar 4.10 dan juga *button* yang ada pada tabel nomor 6 pada gambar yang sama. Gambar 4.11 berikut merupakan rancangan antarmuka *input* dan *edit* data bahan pakan.

Diagram rancangan antarmuka input dan edit data bahan pakan. Elemen-elemen yang ditandai dengan lingkaran bernomor:

1. Kotak input teks "Enter Text" dengan lingkaran 1 di sisi kanan.
2. Tombol dropdown dengan lingkaran 2 di sisi kiri dan ikon panah di sisi kanan.
3. Kotak input teks yang lebih besar "Enter Text" dengan lingkaran 3 di sisi kanan.
4. Tombol dengan lingkaran 4 di sisi kiri.
5. Tombol dengan lingkaran 5 di sisi kanan.

Gambar 4.11 Rancangan antarmuka *input* dan *edit* data bahan pakan

Penjelasan dari rancangan antarmuka pada Gambar 4.11 adalah sebagai berikut,

1. *Text box* untuk memasukkan seri dan nama bahan pakan.
2. *Drop down* untuk memilih golongan bahan pakan yang terdiri dari hijauan dan konsentrat.

3. *Text box* untuk memasukkan kandungan nutrisi bahan pakan yang terdiri dari jumlah BK, jumlah PK, jumlah TDN, dan jumlah NEL beserta harga bahan pakan dan juga nama *user*.
4. *Button* untuk menutup antarmuka.
5. *Button* untuk menyimpan data bahan pakan.



BAB 5 IMPLEMENTASI

Bab ini menjelaskan implementasi sistem berdasarkan analisis kebutuhan dan proses perancangan sistem yang telah dibuat. Pembahasan pada bab ini meliputi penjelasan tentang implementasi algoritma IPSO dalam menyelesaikan masalah optimasi bahan pakan untuk sapi perah, implementasi basis data, dan implementasi antarmuka sistem.

5.1 Implementasi Algoritma

Tahap selanjutnya setelah perancangan ialah mengimplementasikan hasil perancangan sistem optimasi bahan pakan sapi perah menggunakan algoritma IPSO ke dalam kode program. Ada enam proses yang diimplementasi dari sistem optimasi ini diantaranya proses membangun populasi awal, proses menghitung *fitness* partikel, proses menentukan posisi lokal terbaik ($pbest_{id}$), proses menentukan posisi global terbaik ($gbest_d$), proses memperbaiki kecepatan, dan proses memperbaiki posisi partikel.

5.1.1 Implementasi Membangun Populasi Awal

Proses membangun populasi awal merupakan proses inisialisasi partikel. Pada proses ini sejumlah partikel akan dibangkitkan sesuai parameter yang telah dimasukkan *user*. Kode Program 5.1 berikut merupakan implementasi dari proses membangun populasi awal.

Kode Program 5.1 Implementasi membangun populasi awal

```

1  function setPopulasi($ukuran_populasi, $batas_random,
2  $data_pakan_terpilih) {
3      $populasi = array();
4      for ($i = 0; $i < $ukuran_populasi; $i++) {
5          $partikel = array(
6              'pakan' => $this->setPakanPartikel($batas_random,
7              $data_pakan_terpilih),
8              'total_penalty' => 0,
9              'penalty_pbest' => 0,
10             'penalty_gbest' => 0,
11             'total_harga' => 0,
12             'fitness' => 0,
13             'fitness_pbest' => 0,
14             'fitness_gbest' => 0
15             );
16             $populasi[$i] = $partikel;
17         }
18     }
19     return $populasi;
20 }
21
22 function setPakanPartikel ($batas_random, $data_pakan_terpilih) {
23     $pakan = array();
24     $jumlah_pakan = count($data_pakan_terpilih);
25     for ($i = 0; $i < $jumlah_pakan; $i++) {
26         $pakan[$i]['id_bahan_pakan'] =
27         $data_pakan_terpilih[$i]['id_bahan_pakan'];
28         $pakan[$i]['nama_bahan_pakan'] =
29         $data_pakan_terpilih[$i]['nama_bahan_pakan'];
30         $pakan[$i]['bobot_bahan_pakan'] = mt_rand(1 * 100,
31         $batas_random * 100) / 100;
32         $pakan[$i]['kecepatan'] = 0;

```

```

31     $pakan[$i]['pbest'] = 0;
32     $pakan[$i]['gbest'] = 0;
33     }
34     return $pakan;
35 }

```

Penjelasan mengenai Kode Program 5.1 adalah sebagai berikut,

- Baris 1 s.d 18 :
Fungsi membuat sejumlah partikel sesuai variabel ukuran populasi.
- Baris 20 s.d 35 :
Fungsi membuat dimensi pada tiap partikel. Jumlah dimensi disesuaikan dengan variabel data pakan terpilih. Pada tiap dimensi akan mempunyai nilai yang diacak antara 1 sampai nilai variabel batas_random.

5.1.2 Implementasi Menghitung *Fitness*

Setelah mendapatkan populasi awal maka langkah selanjutnya adalah menghitung nilai *fitness* partikel. Perhitungan nilai *fitness* diawali dengan menghitung kandungan nutrisi bahan pakan pada tiap partikel. Kemudian jumlah nutrisi yang tersedia akan dibandingkan dengan kebutuhan nutrisi yang ada. Selisih antara nutrisi yang tersedia dan nutrisi yang dibutuhkan akan membentuk total *penalty*. Selanjutnya total *penalty* beserta total harga pakan pada partikel yang sama dimasukkan ke dalam fungsi *fitness* sehingga didapatkan nilai *fitness*. Implementasi menghitung *fitness* ditunjukkan pada Kode Program 5.2 berikut.

Kode Program 5.2 Implementasi menghitung *fitness*

```

1  function hitungFitnessPartikel($partikel, $data_pakan_terpilih,
2  $kebutuhan, $bobot_harga, $bobot_penalty) {
3      $nutrisi['totalBK'] = $this->hitungBKPakan
4      ($partikel['pakan'], $data_pakan_terpilih);
5      $nutrisi['totalPK'] = $this->hitungPKPakan
6      ($partikel['pakan'], $data_pakan_terpilih);
7      $nutrisi['totalTDN'] = $this->hitungTDNPakan
8      ($partikel['pakan'], $data_pakan_terpilih);
9      $nutrisi['totalNEL'] = $this->hitungNELPakan
10     ($partikel['pakan'], $data_pakan_terpilih);
11     $partikel['total_penalty'] = $this->hitungPenalty
12     ($nutrisi, $kebutuhan);
13     $partikel['total_harga'] = $this->hitungHarga
14     ($partikel['pakan'], $data_pakan_terpilih);
15     $partikel['fitness'] = $this->hitungFungsiFitness
16     ($partikel['total_penalty'], $partikel['total_harga'],
17     $bobot_harga, $bobot_penalty);
18     return $partikel;
19 }
20
21 function hitungFungsiFitness($total_penalty, $total_harga, $w1,
22 $w2) {
23     $fungsi = 100000 / ($w1 * $total_harga + $w2 *
24     $total_penalty * 10000);
25     return round($fungsi, 5);
26 }

```

Penjelasan mengenai Kode Program 5.2 adalah sebagai berikut,

- Baris 3 s.d 10 : proses menghitung jumlah nutrisi pada partikel.
Baris 11 s.d 12 : proses menghitung total *penalty*.

Baris 13 s.d 14 : proses menghitung total harga.

Baris 20 s.d 24 : fungsi menghitung nilai fitness.

5.1.3 Implementasi Menentukan Posisi Lokal Terbaik

Setiap partikel pada algoritma IPSO mempunyai kemampuan untuk menyeleksi nilai *fitness* nya yang terbaik untuk disimpan sebagai posisi lokal terbaik. Hal ini dilakukan sebagai usaha untuk mendekati solusi optimum. Implementasi dari proses ini ditunjukkan pada Kode Program 5.3 berikut.

Kode Program 5.3 Implementasi menentukan posisi lokal terbaik

```

1 function tentukanPbest($old_iterasi, $new_iterasi, $iterasi) {
2     $jumlah_partikel = count($new_iterasi);
3     for ($i = 0; $i < $jumlah_partikel; $i++) {
4         if ($iterasi > 0) {
5             $old_fitness = $old_iterasi[$i]['fitness_pbest'];
6             $new_fitness = $new_iterasi[$i]['fitness'];
7         }
8         if ($iterasi < 1 || $new_fitness > $old_fitness) {
9             $new_iterasi[$i] = $this->tentukanPbestPakan
10            ($new_iterasi[$i], $new_iterasi[$i]);
11            $new_iterasi[$i]['penalty_pbest'] =
12            $new_iterasi[$i]['total_penalty'];
13            $new_iterasi[$i]['fitness_pbest'] =
14            $new_iterasi[$i]['fitness'];
15        }
16    }
17    return $new_iterasi;
18 }

```

Adapun penjelasan dari Kode Program 5.3 adalah sebagai berikut,

- Baris 4 s.d 7 :
Mendapatkan nilai *fitness_pbest* sebelumnya dan nilai *fitness* terbaru jika jumlah iterasi lebih dari 0.
- Baris 8 s.d 15 :
Kondisi jika nilai *fitness_pbest* sebelumnya lebih besar dari nilai *fitness* terbaru maka dilakukan proses *update* posisi lokal terbaik.

5.1.4 Implementasi Menentukan Posisi Global Terbaik

Selain menggunakan posisi lokal terbaik, partikel juga menggunakan posisi global terbaik untuk mendekati solusi optimum. Posisi global terbaik adalah posisi partikel dengan nilai *fitness_pbest* tertinggi dalam satu populasi. Pada tiap iterasi posisi ini akan diperbarui jika didapatkan nilai *fitness_pbest* yang lebih tinggi. Adapun implementasi dari proses ini ditunjukkan pada Kode Program 5.4 berikut.

Kode Program 5.4 Implementasi menentukan posisi global terbaik

```

1 function tentukanGbest($old_iterasi, $new_iterasi, $iterasi) {
2     if ($iterasi > 0) {
3         $old_max = $old_iterasi[0]['fitness_gbest'];
4     }
5     $new_max = $this->getBestFitness($new_iterasi);
6     $jumlah_partikel = count($new_iterasi);
7     if ($iterasi < 1 || $new_max['best_fitness'] > $old_max) {
8         for ($i = 0; $i < $jumlah_partikel; $i++) {
9

```



```

10     $new_iterasi[$i] = $this->tentukanGbestPakan
11     ($new_iterasi[$i], $new_max['best_partikel']);
12     $new_iterasi[$i]['penalty_gbest'] =
13     $new_max['best_partikel']['total_penalty'];
14     $new_iterasi[$i]['fitness_gbest'] =
15     $new_max['best_fitness'];
16     }
17     return $new_iterasi;
18 }

```

Penjelasan dari implementasi menentukan posisi global terbaik pada Kode Program 5.4 adalah sebagai berikut,

- Baris 3 : mendapatkan nilai *fitness_gbest* pada iterasi sebelumnya.
- Baris 5 : mendapatkan nilai *fitness_pbest* maksimum dari seluruh partikel.
- Baris 7 s.d 16 : proses *update* posisi global terbaik jika *fitness_pbest* maksimum Terkini lebih besar dari *fitness_gbest* sebelumnya.

5.1.5 Implementasi Memperbarui Kecepatan Partikel

Proses memperbarui kecepatan partikel melibatkan $pbest_{id}$ dan $gbest_d$ yang telah didapatkan sebelumnya. Kedua komponen tersebut dimasukkan dalam persamaan kecepatan yang telah ditetapkan sehingga didapatkan kecepatan terbaru. Implementasi dari proses ini ditunjukkan pada Kode Program 5.5 berikut.

Kode Program 5.5 Implementasi memperbarui kecepatan partikel

```

1 function updateKecepatan($populasi, $jumlah_iterasi, $iterasi) {
2     $jumlah_partikel = count($populasi);
3     $r1 = mt_rand() / mt_getrandmax();
4     $r2 = mt_rand() / mt_getrandmax();
5     $inertia_weight = $K = null;
6     for ($i = 0; $i < $jumlah_partikel; $i++) {
7         if ($iterasi < $jumlah_iterasi / 2) {
8             $inertia_weight = $this->getInertiaWeight
9             ($populasi[$i], $jumlah_iterasi, $iterasi);
10        } else {
11            $K = $this->getConstrictionFactor($jumlah_iterasi,
12            $iterasi);
13        }
14        $populasi[$i] = $this->hitungKecepatan($populasi[$i],
15        $r1, $r2, $inertia_weight, $K);
16    }
17    return $populasi;
18 }

```

Penjelasan dari Kode Program 5.5 adalah sebagai berikut,

- Baris 3 s.d 4 : mendapatkan angka acak antara 0 sampai 1.
- Baris 7 s.d 10 : kondisi jika iterasi kurang dari jumlah iterasi dibagi 2 maka digunakan komponen *inertia weight*.
- Baris 10 s.d 13: apabila iterasi lebih dari sama dengan jumlah iterasi dibagi 2 maka digunakan komponen *constriction factor*.
- Baris 14 s.d 15: proses *update* kecepatan partikel menggunakan komponen yang telah didapatkan.

5.1.6 Implementasi Memperbarui Posisi Partikel

Setelah kecepatan partikel diperbarui maka selanjutnya posisi partikel juga akan diperbarui. Proses memperbarui posisi partikel diimplementasikan dalam Kode Program 5.6 berikut.

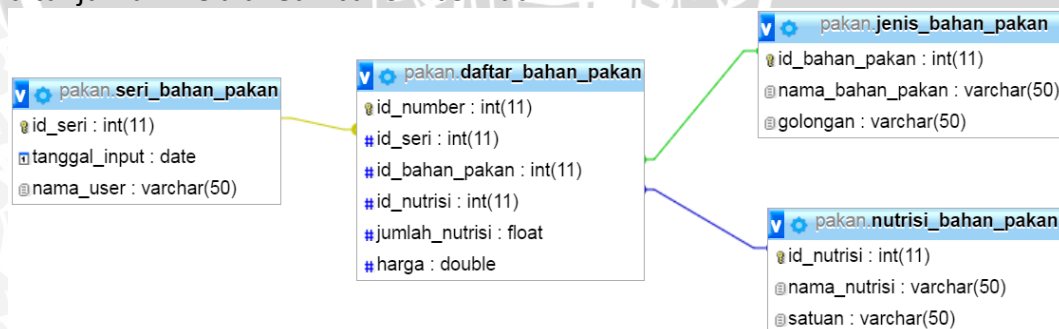
Kode Program 5.6 Implementasi memperbarui posisi partikel

```
1 function updatePosisi($populasi) {
2     $jumlah_partikel = count($populasi);
3     for ($i = 0; $i < $jumlah_partikel; $i++) {
4         $populasi[$i] = $this->
5         updateBototPakan($populasi[$i]);
6     }
7     return $populasi;
8 }
```

Penjelasan dari Kode Program 5.6 adalah sebagai berikut, Baris 3 s.d 6 : merupakan perulangan sejumlah banyaknya partikel untuk melakukan proses *update* posisi partikel.

5.2 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan perancangan basis data. Pada penelitian ini, basis data yang digunakan adalah basis data *MySQL*. Basis data ini berfungsi untuk menyimpan data bahan pakan. Dari hasil implementasi, didapatkan basis data pakan yang terdiri dari empat tabel antara lain tabel daftar_bahan_pakan yang menyimpan kandungan nutrisi pada bahan pakan beserta harga per kilogramnya. Kemudian tabel jenis_bahan_pakan yang berfungsi untuk menyimpan nama bahan pakan. Ketiga adalah tabel nutrisi_bahan_pakan yang menyimpan jenis nutrisi yang diidentifikasi dalam sistem. Dan terakhir adalah tabel seri_bahan_pakan yang menyimpan seri bahan pakan berdasarkan tanggal memasukkan data. Hasil implementasi basis data ditunjukkan melalui Gambar 5.1 berikut.



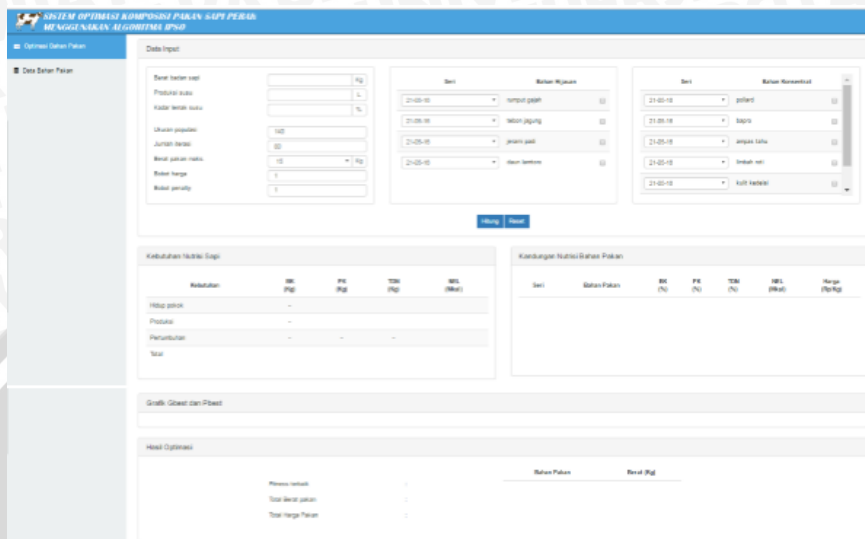
Gambar 5.1 Implementasi basis data

5.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang telah dibuat. Implementasi antarmuka meliputi implementasi antarmuka *input* data, implementasi antarmuka pengelolaan data bahan pakan, dan implementasi antarmuka *input* dan *edit* data bahan pakan.

5.3.1 Implementasi Antarmuka *Input* Data

Antarmuka *input* data merupakan halaman utama sistem. Pada halaman ini proses *input* dan *output* optimasi bahan pakan akan dilakukan. Gambar 5.2 berikut merupakan hasil implementasi antarmuka *input* data.



Gambar 5.2 Implementasi antarmuka *input* data

Proses optimasi diawali dengan memasukkan variabel berat badan sapi, produksi susu, kadar lemak susu, ukuran populasi, jumlah iterasi, batas berat bahan pakan, bobot harga, dan bobot *penalty*. Selain itu *user* juga harus memilih minimal satu bahan pakan hijauan dan satu bahan pakan konsentrat. Kemudian *user* dapat menekan *button* Hitung. Setelah proses optimasi selesai, sistem akan menampilkan tabel kebutuhan nutrisi sapi, tabel bahan pakan yang dipilih *user*, dan hasil perhitungan optimasi. Hasil perhitungan menampilkan *pbest* dan *gbest* pada tiap iterasi dalam bentuk grafik. Hasil optimum dari proses optimasi akan ditampilkan di bagian bawah halaman.

5.3.2 Implementasi Antarmuka Pengelolaan Data Bahan Pakan

Antarmuka pengelolaan data bahan pakan digunakan untuk menampilkan data bahan pakan yang tersimpan dalam basis data. Melalui halaman ini, *user* dapat menambah data bahan pakan baru ke dalam basis data. Selain itu, *user* juga dapat melakukan perubahan pada data yang sudah ada. Adapun hasil implementasi dari antarmuka pengelolaan data bahan pakan ditunjukkan melalui Gambar 5.4 berikut. Untuk menampilkan data bahan pakan, *user* terlebih dulu harus memilih seri bahan pakan pada *drop down* seri. Setelah itu *user* dapat menekan *button* Tampilkan. Maka sistem akan menampilkan data bahan pakan berdasarkan seri yang dipilih *user*. Jika *user* ingin mengubah data bahan pakan, *user* dapat menekan *button* Edit yang tersedia pada setiap baris data bahan pakan. Sedangkan jika ingin menambah data bahan pakan baru, *user* dapat menekan *button* Tambah Bahan Pakan Baru. Ketika *user* menekan *button* Edit atau Tambah Bahan Pakan Baru, sistem akan menampilkan halaman *input* dan *edit* data bahan pakan yang akan dijelaskan pada sub-bab 5.3.3.

Sistem Optimasi Pakan Sapi Perah Menggunakan Algoritma IPSO

Daftar Bahan Pakan

Seri: 28-05-2016

Show: 10 entries Search:

Seri	Golongan	Bahan Pakan	BK (%)	PK (%)	TDN (%)	NEL (Mkal)	Harga (Rp/Kg)	User	Edit
28-05-2016	hijauan	rumpun gajah	36.5	12.8	55.7	1.24465	1250	fitri	<input type="button" value="Edit"/>
28-05-2016	hijauan	tebon jagung	35.1	8.8	68.8	1.5656	1700	fitri	<input type="button" value="Edit"/>
28-05-2016	hijauan	jerami padi	80.8	3.9	45.47	0.994015	500	fitri	<input type="button" value="Edit"/>
28-05-2016	hijauan	daun lamtoro	30	32.12	69.017	1.57092	2000	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	pollard	82.82	18.12	84.78	1.95711	4500	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	bapro	81.79	18.69	32.456	0.675172	3000	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	ampas tahu	20.66	25.69	73.625	1.68381	500	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	limbah roti	68.3	15	89.3	2.06785	250	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	kulit kedelai	90.9	13.9	67.3	1.52885	250	fitri	<input type="button" value="Edit"/>
28-05-2016	konsentrat	bekatul	90.6	15.5	84.8	1.9576	2500	fitri	<input type="button" value="Edit"/>

Showing 1 to 10 of 10 entries

Gambar 5.3 Implementasi antarmuka pengelolaan data bahan pakan

5.3.3 Implementasi Antarmuka *Input* dan *Edit* Data Bahan Pakan

Antarmuka *input* dan *edit* data bahan pakan digunakan untuk menambah data bahan pakan baru maupun mengubah data yang sudah ada. Antarmuka ini akan muncul ketika *user* menekan *button* Tambah Bahan Pakan Baru atau *button* Edit. Gambar 5.4 berikut menunjukkan implementasi antarmuka untuk *input* data bahan pakan baru.

Bahan Pakan

Seri:

Nama Bahan Pakan:

Golongan:

Jumlah BK:

Jumlah PK: %

Jumlah TDN: %

Jumlah NEL: Mkal

Harga: Rp/Kg

Nama User:

Gambar 5.4 Implementasi antarmuka *input* data bahan pakan

Proses *input* data bahan pakan diawali dengan menekan tombol Tambah Bahan Pakan Baru yang dijelaskan pada Gambar 5.3. Setelah itu *user* dapat melengkapi kolom *input* yang tersedia. Kemudian untuk menyimpan data yang telah dimasukkan, *user* dapat menekan *button Save changes* maka data akan tersimpan dalam basis data.

Jika *user* ingin mengubah data yang sudah ada, *user* dapat menekan *button Edit* yang tersedia pada tiap baris data bahan pakan. Data bahan pakan akan ditampilkan ketika *user* sudah memilih seri bahan pakan dan menekan *button Tampilkan* seperti yang dijelaskan pada bab 5.3.3. Gambar 5.5 berikut merupakan implementasi antarmuka untuk *edit* bahan pakan.

Bahan Pakan		
Seri	<input type="text" value="2016-05-28"/>	
Nama Bahan Pakan	<input type="text" value="rumput qajah"/>	
Golongan	<input type="text" value="Hijauan"/>	
Jumlah BK	<input type="text" value="36.5"/>	
Jumlah PK	<input type="text" value="12.8"/>	%
Jumlah TDN	<input type="text" value="55.7"/>	%
Jumlah NEL	<input type="text" value="1.24465"/>	Mkal
Harga	<input type="text" value="1250"/>	Rp/Kg
Nama User	<input type="text" value="fitri"/>	

Gambar 5.5 Implementasi antarmuka *edit* data bahan pakan

Pada proses *edit* data bahan pakan, *user* hanya dapat mengubah jumlah BK, jumlah PK, jumlah TDN, jumlah NEL, dan harga. Selain lima variabel tersebut *user* tidak dapat mengubah data yang telah tersimpan. Untuk menyimpan hasil perubahan, *user* dapat menekan *button Save changes* maka sistem akan melakukan *update* pada basis data.

BAB 6 PENGUJIAN DAN PEMBAHASAN

Bab ini menjelaskan hasil pengujian beserta pembahasan mengenai sistem optimasi bahan pakan sapi perah menggunakan algoritma *improved particle swarm optimization* (IPSO). Pengujian yang dilakukan meliputi pengujian parameter IPSO, pengujian data, dan pengujian perbandingan algoritma.

6.1 Pengujian Parameter IPSO

Pengujian parameter IPSO bertujuan untuk mendapatkan nilai optimum pada parameter yang diuji cobakan sehingga diharapkan dapat menghasilkan solusi yang optimum pula. Adapun parameter yang diuji antara lain jumlah iterasi, ukuran populasi, dan interval berat bahan pakan.

6.1.1. Pengujian Jumlah Iterasi

Pengujian jumlah iterasi digunakan untuk mendapatkan parameter jumlah iterasi yang tepat guna mendapatkan komposisi pakan sapi perah yang optimal. Pengujian dilakukan dengan menghitung rata – rata nilai *fitness* percobaan pada tiap data uji jumlah iterasi. Berikut merupakan rincian variabel yang digunakan untuk menguji parameter jumlah iterasi.

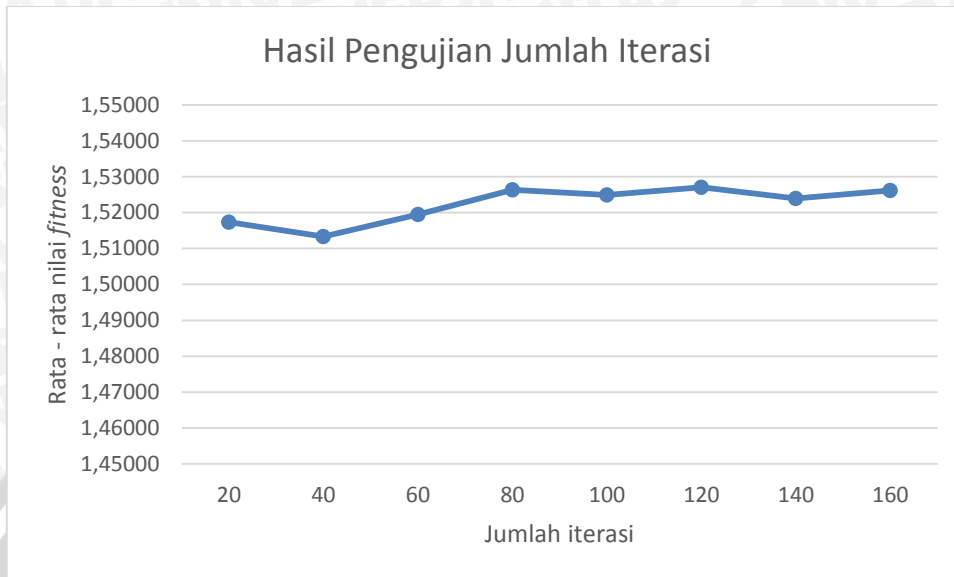
- Berat badan sapi : 450 Kg
- Jumlah produksi susu : 15 L
- Kadar lemak susu : 4,5 %
- Ukuran populasi : 100
- Batas berat maksimum : 15 Kg
- Bahan Pakan :
 - ✓ Rumput gajah
 - ✓ Tebon jagung
 - ✓ Bapro

Pengujian dilakukan sebanyak lima kali pada tiap data uji jumlah iterasi sesuai perancangan yang telah dibuat. Tabel 6.1 berikut merupakan hasil pengujian jumlah iterasi.

Tabel 6.1 Hasil pengujian jumlah iterasi

Jumlah iterasi	Nilai <i>fitness</i> pada percobaan ke-					Rata - rata
	1	2	3	4	5	
20	1,51687	1,50751	1,52607	1,51123	1,52497	1,51733
40	1,50888	1,51584	1,51256	1,52236	1,50722	1,51337
60	1,51979	1,511	1,52435	1,52104	1,52104	1,51944
80	1,52836	1,52736	1,51983	1,52758	1,52867	1,52636
100	1,51233	1,52835	1,52853	1,52689	1,52853	1,52493
120	1,52622	1,52643	1,52754	1,52765	1,5277	1,52711
140	1,52865	1,52766	1,51349	1,52282	1,52718	1,52396
160	1,52864	1,52367	1,5287	1,52241	1,52742	1,52617

Jika rata – rata nilai *fitness* pada Tabel 6.1 ditampilkan dalam grafik maka akan menghasilkan grafik seperti pada Gambar 6.1 berikut.



Gambar 6.1 Grafik hasil pengujian jumlah iterasi

Berdasarkan grafik pada Gambar 6.1, rata – rata nilai *fitness* pada jumlah iterasi 20 dan 40 berkisar 1,51. Kemudian meningkat pada jumlah iterasi 60. Dan terus meningkat sampai jumlah iterasi 80. Dari penjelasan tersebut diketahui bahwa jumlah iterasi berpengaruh terhadap nilai *fitness* yang dihasilkan. Jumlah iterasi yang terlalu sedikit dapat menimbulkan konvergensi dini karena area penelurusan partikel terbatas. Sebaliknya jika jumlah iterasi terlalu banyak akan membuat tingkat komputasi semakin meningkat. Padahal nilai *fitness* yang dihasilkan tidak ada kenaikan yang signifikan (Engelbrecht, 2007). Oleh karena itu dapat diambil kesimpulan bahwa jumlah iterasi optimum adalah 80. Hal itu disebabkan rata – rata nilai *fitness* dari jumlah iterasi 80 sampai 160 terlihat stabil pada kisaran 1,52.

6.1.2. Pengujian Ukuran Populasi

Pengujian ukuran populasi digunakan untuk mendapatkan parameter ukuran populasi yang tepat guna mendapatkan komposisi pakan sapi perah yang optimal. Sama seperti pengujian jumlah iterasi, pengujian ini juga dilaksanakan dengan menghitung rata – rata nilai *fitness* percobaan pada tiap kategori ukuran populasi. Berdasarkan hasil pengujian jumlah iterasi didapatkan jumlah iterasi optimum adalah 80 maka pada rincian variabel berikut digunakan jumlah iterasi 80.

- Berat badan sapi : 450 Kg
- Jumlah produksi susu : 15 L
- Kadar lemak susu : 4,5 %
- Jumlah iterasi (T_{max}) : 80
- Batas berat maksimum : 15 Kg
- Bahan Pakan :

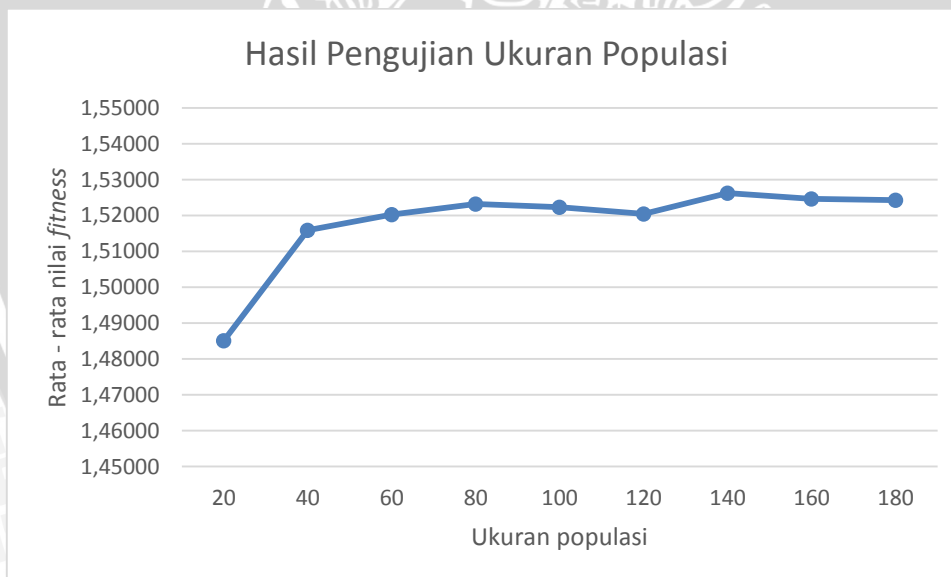
- ✓ Rumput gajah
- ✓ Tebon jagung
- ✓ Bapro

Berdasarkan perancangan yang telah dibuat, pengujian ukuran populasi dilakukan sebanyak lima kali pada tiap data uji ukuran populasi. Hasil dari pengujian ini ditunjukkan pada Tabel 6.2 berikut.

Tabel 6.2 Hasil pengujian ukuran populasi

Ukuran populasi	Nilai <i>fitness</i> pada percobaan ke-					Rata - rata
	1	2	3	4	5	
20	1,51253	1,5193	1,43488	1,47901	1,47982	1,48511
40	1,522	1,52712	1,515	1,51821	1,49706	1,51588
60	1,52383	1,52295	1,52772	1,50707	1,51948	1,52021
80	1,51566	1,52464	1,52208	1,5274	1,52618	1,52319
100	1,52719	1,52298	1,51896	1,52137	1,52091	1,52228
120	1,52287	1,5076	1,52847	1,52542	1,51771	1,52041
140	1,52753	1,52736	1,52816	1,52119	1,52671	1,52619
160	1,52549	1,5277	1,51736	1,52782	1,52465	1,52460
180	1,52812	1,5272	1,52654	1,52821	1,5114	1,52429

Grafik hasil pengujian ukuran populasi terhadap rata – rata nilai *fitness* ditunjukkan melalui Gambar 6.2 berikut.



Gambar 6.2 Grafik hasil pengujian ukuran populasi

Berdasarkan grafik pada Gambar 6.2, rata – rata nilai *fitness* meningkat cukup drastis dari ukuran populasi 20 ke ukuran populasi 40. Kemudian meningkat menjadi 1,52 pada ukuran populasi 60. Dan terus meningkat sampai pada ukuran populasi 80. Setelah itu rata – rata nilai *fitness* turun secara perlahan sampai ukuran populasi 120. Kemudian naik lagi pada ukuran populasi 140. Setelah itu turun lagi sedikit pada iterasi 160 lalu stabil sampai ukuran populasi 180. Gejala naik turun ini sebenarnya berada pada kisaran 1,52 yang dimulai dari ukuran

populasi 60. Akan tetapi rata – rata nilai *fitness* mencapai maksimum pada ukuran populasi 140. Oleh karena itu, diambil kesimpulan bahwa ukuran populasi yang optimal adalah 140.

Hasil pengujian ini menunjukkan bahwa ukuran populasi berpengaruh terhadap nilai *fitness* yang dihasilkan. Ukuran populasi yang besar akan meningkatkan keanekaragaman partikel dalam populasi. Hal ini membuat area penelurusan partikel semakin luas sehingga kemungkinan untuk mencapai solusi optimum semakin besar. Akan tetapi ukuran populasi yang terlalu besar akan berdampak pada tingkat komputasi yang semakin kompleks (Engelbrecht, 2007).

6.1.3. Pengujian Interval Berat Bahan Pakan

Pengujian interval berat bahan pakan bertujuan untuk mendapatkan interval berat bahan pakan yang tepat. Interval ini digunakan sebagai batasan dalam mengacak posisi awal partikel pada proses inialisasi populasi. Pengujian ini juga dilakukan dengan menghitung rata – rata nilai *fitness* percobaan pada tiap kategori interval. Hasil pengujian parameter sebelumnya digunakan pada rincian variabel pengujian berikut.

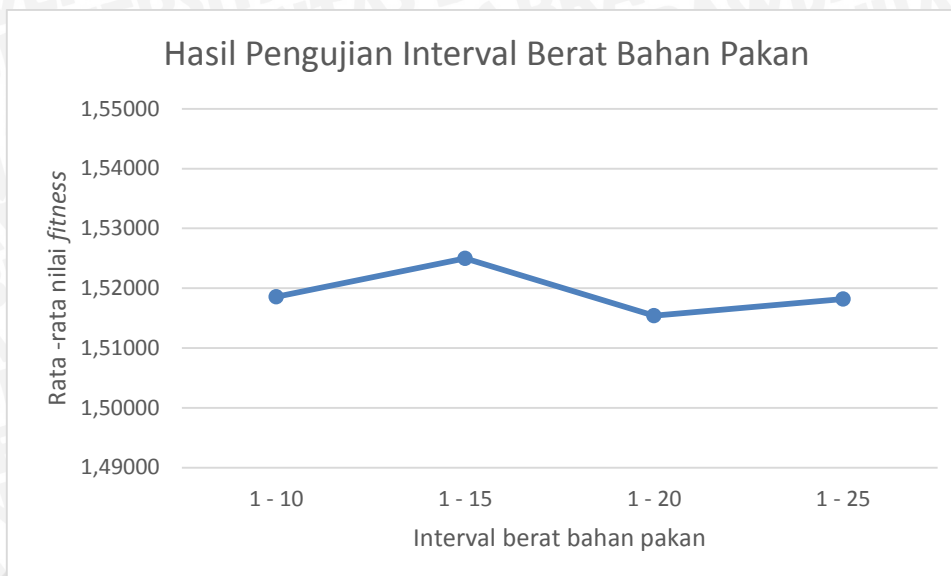
- Berat badan sapi : 450 Kg
- Jumlah produksi susu : 15 L
- Kadar lemak susu : 4.5 %
- Jumlah iterasi : 80
- Ukuran populasi : 140
- Bahan Pakan :
 - ✓ Rumput gajah
 - ✓ Tebon jagung
 - ✓ Bapro

Berdasarkan perancangan pengujian interval berat bahan pakan, pengujian ini dilakukan sebanyak lima kali pada tiap data uji interval berat bahan pakan. Hasil dari pengujian ini ditampilkan pada Tabel 6.3 berikut.

Tabel 6.3 Hasil pengujian interval berat bahan pakan

Interval	Nilai <i>fitness</i> pada percobaan ke-					Rata - rata
	1	2	3	4	5	
1 - 10	1,5277	1,52661	1,49811	1,52866	1,51167	1,51855
1 - 15	1,52701	1,52818	1,52497	1,51966	1,5251	1,52498
1 - 20	1,5229	1,4907	1,52212	1,51507	1,52623	1,51540
1 - 25	1,52741	1,51422	1,52563	1,52251	1,50128	1,51821

Dari hasil pengujian pada Tabel 6.3 kemudian dibentuk grafik terhadap rata – rata nilai *fitness* seperti pada Gambar 6.3 berikut.



Gambar 6.3 Grafik hasil pengujian interval berat bahan pakan

Berdasarkan grafik pada Gambar 6.3 diketahui bahwa rata – rata nilai *fitness* yang dihasilkan ada pada kisaran 1,51 sampai 1,52, atau lebih tepatnya berada pada rata – rata 1,51929. Namun rata – rata nilai *fitness* maksimum dicapai ketika interval bobot bahan antara 1 sampai 15. Oleh karena itu, interval berat bahan pakan yang optimal adalah 1 sampai 15.

Hasil pengujian ini membuktikan bahwa interval berat bahan pakan berpengaruh terhadap nilai *fitness* yang dihasilkan. Interval yang terlalu kecil mengakibatkan partikel membutuhkan jumlah iterasi yang cukup lama karena jaraknya yang terlalu jauh dengan solusi optimum. Akan jika intervalnya terlalu besar bisa mengakibatkan posisi awal partikel berada di luar area solusi optimum.

6.2 Pengujian Data

Pengujian data bertujuan untuk mengukur kualitas solusi yang dihasilkan menggunakan algoritma IPSO dengan data hasil observasi di lapangan. Proses observasi dilakukan di desa Dresel, kecamatan Batu, kota Batu. Hasil observasi dari beberapa peternak di desa Dresel ditampilkan dalam Tabel 6.4 berikut.

Tabel 6.4 hasil observasi di desa Dresel, kecamatan Batu, kota Batu

No.	Keadaan Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
1	BB : 400 kg Produksi : 8 L Lemak : 4 %	Rumput gajah	1250	30	37500
		Bapro	3000	3	9000
Jumlah				33	46500
2	BB : 400 kg Produksi : 9 L Lemak : 4,2 %	Rumput gajah	1250	25	31250
		Pollard	4500	4	18000
Jumlah				29	49250
3	BB : 380 kg	Rumput gajah	1250	25	31250

No.	Keadaan Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
	Produksi : 6 L Lemak : 4,2 %	Pollard	4500	2,5	11250
Jumlah				27,5	42500
4	BB : 400 kg Produksi : 8 L Lemak : 4.2 %	Rumput gajah	1250	25	31250
		Pollard	4500	3	13500
Jumlah				28	44750
5	BB : 400 kg Produksi : 8 L Lemak : 4 %	Rumput gajah	1250	30	37500
		Pollard	4500	3	13500
Jumlah				33	51000

Kemudian peneliti melakukan pengujian menggunakan kondisi sapi yang sama seperti pada Tabel 6.4. Pengujian dilakukan dengan menjalankan sistem menggunakan nilai parameter optimum berdasarkan hasil pengujian parameter IPSO. Hasil dari pengujian data menggunakan sistem ditampilkan pada Tabel 6.5 berikut.

Tabel 6.5 Hasil dengan sistem optimasi komposisi pakan sapi perah menggunakan IPSO

No.	Keadaan Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
1	BB : 400 kg Produksi : 8 L Lemak : 4 %	Rumput gajah	1250	6,77	8462,5
		Bapro	3000	7,95	23850
Jumlah				14,72	32312,5
2	BB : 400 kg Produksi : 9 L Lemak : 4,2 %	Rumput gajah	1250	1,04	1300
		Pollard	4500	6,84	30780
Jumlah				7,88	32080
3	BB : 380 kg Produksi : 6 L Lemak : 4,2 %	Rumput gajah	1250	1,02	1275
		Pollard	4500	5,54	24930
Jumlah				6,56	26205
4	BB : 400 kg Produksi : 8 L Lemak : 4,2 %	Rumput gajah	1250	1,03	1287,5
		Pollard	4500	6,46	29070
Jumlah				7,49	30357,5
5	BB : 400 kg Produksi : 8 L Lemak : 4 %	Rumput gajah	1250	1,21	1512,5
		Pollard	4500	6,28	28260
Jumlah				7,49	29772,5

Perbandingan total harga pakan antara data hasil observasi dengan hasil menggunakan sistem disajikan dalam Tabel 6.6 berikut.

Tabel 6.6 Hasil pengujian data

No	Total Harga Pakan (Rp)		Selisih (Rp)
	Data Hasil Observasi	IPSO	
1	46500	32312,5	14187,5
2	49250	32080	17170
3	42500	26205	16295
4	44750	30357,5	14392,5
5	51000	29772,5	21227,5
Rata – rata (Rp)			16654,5

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.6, sistem optimasi berbasis algoritma IPSO dapat menghasilkan komposisi pakan 35% lebih murah dari pada cara konvensional yang dilakukan oleh peternak sapi di desa Dresel, Batu. Dengan menggunakan sistem ini, peternak dapat menghemat biaya pengeluaran pakan sebesar Rp 16654,5 pada setiap pemberian pakan.

6.3 Pengujian Perbandingan Algoritma

Pengujian perbandingan algoritma bertujuan untuk mengukur kualitas solusi yang dihasilkan menggunakan algoritma IPSO solusi yang dihasilkan menggunakan algoritma lainnya. Pada penelitian ini, algoritma yang digunakan sebagai pembanding adalah algoritma PSO. Penelitian yang dirujuk adalah hasil penelitian Khaqqo (2016) tentang optimasi bahan pakan sapi perah menggunakan algoritma PSO. Berdasarkan pengujian yang dilakukan oleh Khaqqo (2016) didapatkan hasil optimasi bahan pakan seperti pada Tabel 6.4 berikut.

Tabel 6.7 Hasil optimasi bahan pakan sapi perah menggunakan algoritma PSO

No.	Variabel Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
1	BB : 400 kg Produksi : 8.8 L Lemak : 4 %	Rumput gajah	1250	7.1	8875
		Bapro	3000	9.7	29100
		Jumlah		16.8	37975
2	BB : 400 kg Produksi : 9.1 L Lemak : 4.2 %	Rumput gajah	1250	5.3	6625
		Pollard	4500	9.1	40950
		Jumlah		14.4	47575
3	BB : 380 kg Produksi : 6.8 L Lemak : 4.2 %	Rumput gajah	1250	13	16250
		Bapro	3000	8	24000
		Jumlah		21	40250
4	BB : 400 kg Produksi : 8.4 L Lemak : 4.2 %	Rumput gajah	1250	8.7	10875
		Bapro	3000	9.9	29700
		Jumlah		18.6	40575

No.	Variabel Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
5	BB : 400 kg Produksi : 8.8 L Lemak : 4 %	Rumput gajah	1250	5.6	7000
		Bapro	3000	11.4	34200
Jumlah				17	41200

Dengan variabel sapi yang sama, peneliti melakukan pengujian menggunakan algoritma IPSO. Dari pengujian yang dilakukan didapatkan hasil seperti pada Tabel 6.5 berikut.

Tabel 6.8 Hasil optimasi bahan pakan sapi perah menggunakan algoritma IPSO

No.	Variabel Sapi	Bahan Pakan	Harga / kg Bahan Pakan (Rp)	Jumlah (kg)	Total Harga (Rp)
1	BB : 400 kg Produksi : 8.8 L Lemak : 4 %	Rumput gajah	1250	6,96	8700
		Bapro	3000	8,47	25410
Jumlah				15,43	34110
2	BB : 400 kg Produksi : 9.1 L Lemak : 4.2 %	Rumput gajah	1250	1,01	1262,5
		Pollard	4500	6,78	30510
Jumlah				7,79	31772,4
3	BB : 380 kg Produksi : 6.8 L Lemak : 4.2 %	Rumput gajah	1250	5,65	7062,5
		Bapro	3000	8,41	25230
Jumlah				14,06	32292,5
4	BB : 400 kg Produksi : 8.4 L Lemak : 4.2 %	Rumput gajah	1250	6,89	8612,5
		Bapro	3000	8,37	25110
Jumlah				15,26	33722,5
5	BB : 400 kg Produksi : 8.8 L Lemak : 4 %	Rumput gajah	1250	6,97	8712,5
		Bapro	3000	8,46	25380
Jumlah				15,43	34092,5

Kemudian dari kedua hasil optimasi dibandingkan dari segi total harga pakan seperti pada Tabel 6.6 berikut.

Tabel 6.9 Hasil pengujian perbandingan algoritma

No	Total Harga Pakan (Rp)		Selisih (Rp)
	PSO	IPSO	
1	37975	34110	3865
2	47575	31772,4	15802,6
3	40250	32292,5	7957,5
4	40575	33722,5	6852,5
5	41200	34092,5	7107,5
Rata – rata (Rp)			8317

Dari Tabel 6.9 diketahui bahwa algoritma IPSO dapat menghasilkan komposisi bahan pakan yang lebih murah dibandingkan menggunakan algoritma PSO.

Dengan sistem optimasi bahan pakan menggunakan algoritma IPSO, peternak dapat menghemat uang sebesar Rp 8317 pada tiap pemberian pakan.

Selain itu masing – masing algoritma menggunakan nilai parameter optimalnya dalam melakukan pengujian di atas. Berdasarkan hasil pengujian pada penelitian Khaqqo (2016) disebutkan bahwa ukuran populasi optimum pada PSO untuk menyelesaikan masalah ini adalah 400, dengan jumlah iterasi optimum sebesar 350. Sedangkan pada algoritma IPSO, ukuran populasi optimum adalah 140 dan jumlah iterasi optimum adalah 80. Dengan begitu waktu komputasi algoritma IPSO akan lebih cepat dari pada algoritma PSO. Maka dapat disimpulkan bahwa algoritma IPSO lebih efisien dari pada algoritma PSO dalam menghasilkan solusi optimum.

6.4 Pembahasan Hasil Pengujian

Dari proses pengujian parameter IPSO yang telah dilakukan, diperoleh nilai optimum pada tiap parameter yang diuji. Pengujian parameter IPSO meliputi pengujian ukuran populasi yang menghasilkan nilai optimum 80, kemudian pengujian jumlah iterasi yang memperoleh nilai optimum 140, dan pengujian interval berat bahan pakan yang menunjukkan nilai optimum pada interval 1 sampai 15. Dengan menjalankan sistem menggunakan nilai parameter optimal didapatkan nilai *fitness* tertinggi sebesar 1,52818.

Berdasarkan hasil pengujian data, sistem optimasi berbasis algoritma IPSO dapat menghasilkan komposisi pakan yang lebih murah dari pada cara konvensional. Dari pengujian tersebut, sistem dapat menghasilkan komposisi pakan 35% lebih murah dibandingkan cara konvensional. Dengan sistem ini, peternak dapat memperoleh rekomendasi pakan yang telah memenuhi kebutuhan nutrisi sapi, namun dengan biaya yang lebih hemat sebesar Rp 16654,5 pada setiap pemberian pakan.

Selain itu dari hasil pengujian perbandingan algoritma, sistem optimasi menggunakan algoritma IPSO masih menghasilkan komposisi pakan yang lebih murah dibandingkan dengan sistem optimasi berbasis PSO. Dibandingkan dengan algoritma PSO, sistem optimasi menggunakan algoritma IPSO dapat menghemat biaya pengeluaran pakan sebesar Rp 8317 pada tiap pemberian pakan. Algoritma IPSO juga terbukti lebih efisien dalam mendapatkan solusi optimum dari pada algoritma PSO.

Solusi yang diharapkan dari sistem ini adalah komposisi bahan pakan yang optimal, yaitu komposisi pakan yang memenuhi kebutuhan sapi dan memiliki total harga pakan yang seminimal mungkin. Berdasarkan hasil pengujian yang didapatkan, sistem optimasi menggunakan algoritma IPSO dianggap mampu untuk memberikan rekomendasi komposisi pakan yang optimal. Meskipun dari hasil pengujian menunjukkan bahwa sistem ini menghasilkan komposisi pakan yang lebih murah dari pada cara konvensional, namun hasil keluaran dari sistem belum diujikan secara langsung terhadap ternak sapi perah.

BAB 7 PENUTUP

7.1 Kesimpulan

1. Algoritma *improved particle swarm optimization* (IPSO) dapat digunakan untuk menyelesaikan masalah optimasi komposisi pakan sapi perah. Proses penyelesaian masalah menggunakan IPSO dimulai dengan membangun populasi awal. Kemudian perhitungan *fitness* untuk mengukur kualitas solusi pada tiap partikel. Setelah itu penentuan posisi lokal terbaik (*pbest*) dan posisi global terbaik (*gbest*). Kemudian memperbaiki kecepatan dan posisi partikel. Serangkaian proses tersebut merupakan upaya untuk mendekati solusi optimum. Solusi yang diharapkan adalah komposisi pakan sapi perah yang mencukupi kebutuhan nutrisi sapi dengan harga pakan yang minimal.
2. Pengujian parameter IPSO bertujuan untuk menentukan parameter yang tepat guna mendapatkan komposisi bahan pakan sapi perah yang optimal. Berdasarkan hasil pengujian, parameter jumlah iterasi yang optimum adalah 80. Sedangkan untuk ukuran populasi optimum adalah 140 dan interval bobot bahan pakan optimum adalah 1 sampai 15.
3. Untuk mengetahui kualitas yang dihasilkan menggunakan IPSO maka dilakukan pengujian data dan pengujian perbandingan algoritma. Dari pengujian data diketahui bahwa sistem dapat menghasilkan komposisi pakan yang lebih murah sebesar Rp 16654 dari pada cara konvensional. Sedangkan dari hasil pengujian perbandingan algoritma diketahui bahwa algoritma IPSO dapat menghasilkan komposisi pakan Rp 8317 lebih murah dari pada menggunakan PSO. Selain itu algoritma IPSO terbukti lebih efisien dibandingkan algoritma PSO dalam mendapatkan solusi optimum.

7.2 Saran

1. Ada berbagai macam bentuk pengembangan algoritma PSO. Maka pada penelitian selanjutnya dapat dikembangkan algoritma PSO menjadi algoritma IPSO yang lain atau menggunakan algoritma IPSO lainnya yang telah ada untuk menyelesaikan masalah optimasi komposisi pakan sapi perah. Dengan demikian peneliti selanjutnya dapat mengetahui algoritma mana yang lebih tepat untuk menyelesaikan masalah ini.
2. Nutrisi lainnya selain bahan kering (BK), protein kasar (PK), *total digestible nutrient* (TDN), dan *net energy of lactation* (NE_L) bisa ditambahkan pada penelitian mendatang supaya komposisi pakan yang dihasilkan lebih optimum.

DAFTAR PUSTAKA

- Aggraeni, A., 2012. Perbaikan Genetik Sifat Produksi Susu dan Kualitas Susu Sapi Friesen Holstein Melalui Seleksi. *WARTAZOA*, 22(1), pp. 1-11.
- Ako, A., 2011. *Produksi Ternak Perah*, Makassar: Universitas Hasanuddin.
- Alfian, A., 2014. *Manajemen Pemberian Pakan Sapi Perah Laktasi di Peternakan Anggota KUD Batu, Kota Batu*, Malang: Program Studi Peternakan, Fakultas Peternakan, Universitas Brawijaya.
- Bamualin, A. M., Kusmartono & Kuswandi, 2009. Aspek Nutrisi Sapi Perah. Dalam: K. A. Santoso, K. Diwyanto & T. Toharmat, penyunt. *Profil Usaha Peternakan Sapi Perah Di Indonesia*. Bogor: LIPI Press, pp. 165-208.
- Chen, S.-Y., Hung, Y.-H., Wu, C.-H. & Huang, S.-T., 2015. Optimal energy management of a hybrid electric powertrain system. *Applied Energy*, Volume 160, pp. 132-145.
- Council, N. R., 2001. *Nutrient Requirements of Dairy Cattle Seventh Revised Edition, 2001*, Washington, D.C.: National Academy Press.
- Engelbrecht, A. P., 2007. *Computational Intelligence: An Introduction*. 2nd penyunt. South Africa: John Wiley & Sons Ltd.
- Fakhiroh, D., Mahmudy, W. F. & Indriati, 2015. Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Genetika. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(14).
- Jayanegara, A., 2014. Evaluasi Pemberian Pakan Sapi Perah Laktasi Menggunakan Standar NRC 2001: Studi Kasus Peternakan Di Sukabumi. *Dalam: Prosiding Seminar Nasional, Optimalisasi Sumber Daya Lokal Pada Peternakan Rakyat Berbasis Teknologi, Universitas Hasanuddin, Makassar*, pp. 86-91.
- Khaqqo, A., 2016. *Optimasi Komposisi Pakan Sapi Perah Menggunakan Algoritma Particle Swarm Optimization (PSO)*, Malang: Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.
- Lu, Y., Liang, M., Ye, Z. & Chao, L., 2015. Improved particle swarm optimization algorithm and its application in text feature selection. *Applied Soft Computing*, Volume 35, pp. 629-636.
- Mahmudy, W. F., 2015. Improved Particle Swarm Optimization untuk Menyelesaikan Permasalahan Part Type Selection dan Machine Loading pada Flexible Manufacturing System (FMS). *Dalam: Konferensi Nasional Sistem Informasi, Universitas Klabat, Airmadidi, Minahasa Utara, Sulawesi Utara*, pp. 1003 - 1008.
- Milah, H. & Mahmudy, W. F., 2015. Implementasi Algoritma Evolution Strategies Untuk Optimasi Komposisi Pakan Ternak Sapi Potong. *DORO: Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, 5(11).

- Pan, D. & Liu, Z., 2011. An Improved Particle Swarm Optimization Algorithm. *Emerging Research in Artificial Intelligence and Computational Intelligence*, Volume 237, pp. 550-556.
- Tuegeh, M., Soeprijanto & Purnomo, M. H., 2009. Modified Improved Particle Swarm Optimization for Optimal Generator Scheduling. *Dalam: Seminar Nasional Aplikasi Teknologi Informasi 2009 (SNATI 2009)*, Yogyakarta, pp. 85 - 90.
- Umiyasih, U. & Anggraeny, Y. N., 2008. Strategi Pakan Alternatif Sebagai Upaya Efisiensi Pada Usaha Sapi Perah Rakyat. *Dalam: Semiloka Nasional Prospek Industri Sapi Perah Menuju Perdagangan Bebas - 2020, Loka Penelitian Sapi Potong, Jawa Timur*, pp. 226 - 231.
- Wu, P., Gao, L., Zou, D. & Li, S., 2011. An improved particle swarm optimization algorithm for reliability problems. *ISA Transactions*, Volume 50, pp. 71-81.
- Yusdja, Y., 2005. Kebijakan Ekonomi Industri Agribisnis Sapi Perah di Indonesia. *Analisis Kebijakan Pertanian*, 3(3), pp. 257-268.



LAMPIRAN A HASIL WAWANCARA NARASUMBER

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

Pertanyaan :

Bagaimanakah cara yang baik dalam menentukan kebutuhan nutrisi sapi perah?

Jawaban :

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

Pertanyaan :

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

Jawaban :

Kemampuan sapi berproduksi ditentukan oleh genetiknya. Sedangkan pakan dan lingkungan hanya mendukung potensi genetik sapi yang artinya dapat memaksimalkan potensi produksi.

Pertanyaan :

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

Jawaban :

Tidak ada penentuan komposisi pakan minimal untuk setiap sapi. Kembali lagi untuk menentukan kebutuhan pakan harus dihitung terlebih dahulu kebutuhan nutrisinya. Dari data berat badan, produksi, dan kandungan lemak susu sapi dapat ditentukan berapa kebutuhan tiap nutrisinya. Kemudian kebutuhan nutrisi tersebut ditotal dan barulah dari total nutrisi tersebut dapat dihitung kebutuhan pakannya berapa. Jenis pakan sapi perah harus mengandung hijauan dan konsentrat.

LAMPIRAN B HASIL OBSERVASI DATA PENERIMAAN SUSU DI KUD BATU DAN DATA PEMBERIAN PAKAN PETERNAK DI DESA DRESEL, KECAMATAN BATU, KOTA BATU

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

Maskur Afandi (664)

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

Dosan (675)

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

2	16-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L
1	Kamis, 17-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	9 L
		S. Rumput gajah	25 kg	Pollard	4 kg	5 L
		P. Rumput gajah	25 kg	Pollard	4 kg	8 L
2	17-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
		S. Rumput gajah	25 kg	Pollard	4 kg	5 L
1	Jumat, 18-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	8 L
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
2	18-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	5 L
		P. Rumput gajah	25 kg	Pollard	4 kg	8 L
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L
1	Sabtu, 19-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	9 L
		S. Rumput gajah	25 kg	Pollard	4 kg	5 L
		P. Rumput gajah	25 kg	Pollard	4 kg	8 L
2	19-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
		S. Rumput gajah	25 kg	Pollard	4 kg	5 L
1	Minggu, 20-12-2015	P. Rumput gajah	25 kg	Pollard	4 kg	8 L
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L
		P. Rumput gajah	25 kg	Pollard	4 kg	9 L
2	20-12-2015	S. Rumput gajah	25 kg	Pollard	4 kg	5 L
		P. Rumput gajah	25 kg	Pollard	4 kg	8 L
		S. Rumput gajah	25 kg	Pollard	4 kg	4 L

Kartubi (680)

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

Siono (685)

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

	19-12-2015	S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Minggu, 20-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	
1	Senin, 21-12-2015	P. Rumput gajah	25 kg	Pollard	3 kg	8 L	
		S. Rumput gajah	25 kg	Pollard	3 kg	5 L	

Eko (2584)

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

