PEMODELAN KOMPOSISI PAKAN KAMBING PERANAKAN ETAWA (PE) UNTUK OPTIMASI KANDUNGAN GIZI SUSU MENGGUNAKAN *ARTIFICIAL NEURAL NETWORK* (ANN) DAN *PARTICLE SWARM OPTIMIZATION* (PSO)

(Studi Kasus pada UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Canny Amerilyse Caesar
NIM: 125150201111023

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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

PEMODELAN KOMPOSISI PAKAN KAMBING PERANAKAN ETAWA (PE) UNTUK OPTIMASI KANDUNGAN GIZI SUSU MENGGUNAKAN *ARTIFICIAL NEURAL NETWORK* (ANN) DAN *PARTICLE SWARM OPTIMIZATION* (PSO)
(Studi Kasus pada UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Canny Amerilyse Caesar
NIM: 125150201111023

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Juli 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Imam Cholissodin, S.Si, M.Kom	M. Ali Fauzi, S.Kom, M.Kom
NIK: 201201 850719 1 001	NIK: 201502 890101 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, 28 Juli 2016

Canny Amerilyse Caesar

NIM: 125150201111023

KATA PENGANTAR

Puji syukur kehadirat Tuhan YME yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga laporan skripsi yang berjudul **"Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) untuk Optimasi Kandungan Gizi Susu Menggunakan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO) (Studi Kasus pada UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang)"** ini dapat terselesaikan dengan baik. Skripsi ini disusun untuk memenuhi syarat memperoleh gelar Sarjana Komputer pada Program Studi Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Imam Cholissodin, S.Si, M.Kom selaku dosen pembimbing skripsi I yang telah menyediakan waktu, tenaga dan pemikirannya untuk membimbing, berdiskusi dengan penulis dan membantu penyelesaian penelitian ini dengan baik.
2. Bapak M. Ali Fauzi, S.Kom, M.Kom selaku dosen pembimbing skripsi II yang telah menyediakan waktu dan memberikan banyak masukan dan motivasi kepada penulis sehingga skripsi ini dapat terselesaikan dengan baik.
3. Bapak Rokhim, S.Pt, Ibu drh. Jayawulandari dan Bapak Kayat, S.Pt selaku pakar di bidang pakan, kesehatan hewan dan hasil uji laboratorium kandungan gizi susu kambing PE di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang yang telah bersedia menjadi narasumber serta membimbing penulis dalam pengerjaan skripsi ini.
4. Bapak Budi Darma Setiawan, S.Kom, M.Cs dan Bapak Issa Arwani, S.Kom, M.Sc selaku dosen penguji 1 dan 2 yang telah memberikan saran perbaikan skripsi ini.
5. Kedua orangtua penulis, kakak, dan seluruh keluarga besar atas segala nasehat, motivasi, kasih sayang, perhatian, kesabarannya dalam membesarkan dan mendidik penulis serta yang selalu senantiasa mendukung dalam doa dan semangat demi terselesaikannya skripsi ini.
6. Seluruh dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini
7. Sahabat penulis Martua Indah Christian Saragih Napitu, Latifah Hanum, Yuni Widyaningtyas, Edelin bernadifta, Uly Theresia Tambunan, Yuanita Muharyani, Rizky Salsabella, Zahra Swastika Putri, Novinda Fiqih, Uswatun Hasanah, dan Sri Retno Mia yang selalu menemani, bertukar ilmu, semangat dan motivasi kepada penulis selama masa studi dan penyusunan skripsi.

8. Teman-teman Daniel angkatan 2012 yang selalu menghibur, memberikan motivasi dan semangat serta rasa kebersamaan kepada penulis selama ini.
9. Seluruh keluarga besar Daniel terimakasih atas kebersamaan, keceriaan, saran dan dorongan kepada penulis selama masa studi.
10. Pihak-pihak lain yang tidak dapat penulis sebut satu persatu, yang turut membantu penyelesaian skripsi baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan. Oleh karena itu, penulis bersedia menerima kritik dan saran yang membangun demi memperbaiki diri. Harapan penulis semoga skripsi ini dapat memberikan manfaat pada semua pihak, terutama bagi penulis dan pengembang yang akan mengembangkan penelitian dari penulis. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, Juli 2016

Penulis

camerilyse@outlook.com

ABSTRAK

Susu merupakan salah satu sumber protein hewani yang mengandung semua zat yang dibutuhkan tubuh. Ternak penghasil susu utama di Indonesia yaitu sapi perah, namun produksi susunya belum dapat mencukupi kebutuhan masyarakat. Alternatifnya adalah kambing peranakan etawa (PE). Tingginya kualitas kandungan gizi susu sangat dipengaruhi oleh beberapa faktor salah satunya, yaitu faktor pakan. Bagian peternakan kambing PE di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang masih menghadapi permasalahan, yaitu rendahnya kemampuan dalam memberikan komposisi pakan terhadap kambing PE. Kekurangan tersebut berpengaruh terhadap kualitas susu yang dihasilkan. Diperlukan pengetahuan rekayasa kandungan gizi susu untuk menentukan komposisi pakan dalam menghasilkan susu premium dengan kandungan gizi optimal. Penulis menggunakan metode *Artificial neural network* (ANN) dan *Particle Swarm Optimization* (PSO) untuk membuat pemodelan pakan kambing dalam mengoptimasi kandungan gizi susu kambing. Dalam analisa pengujian yang dilakukan dengan kasus untuk berat badan kambing 32 kg, serta jenis pakan yang digunakan yaitu rumput Odot 70% dan rumput Raja 30% dapat menaikkan protein hingga 3.42% dan menurunkan lemak hingga 5.19% .

Kata Kunci: Susu Kambing, Optimasi, Artificial Neural Network (ANN), *Particle Swarm Optimization* (PSO), Kandungan nutrisi pakan.

ABSTRACT

Milk is one of the animal protein sources which it contains all of the substances needed by human body. The main milk producer cattle in Indonesia is dairy cow, however its milk production has not fulfilled the society needs. The alternative is the goat, the Etawa crossbreed (PE). The high quality of milk nutrients content is greatly influenced by some factors one of them, is the food factor. The PE goat livestock division of the UPT Cattle Breeding and the Cattle Food Greenery in Singosari-Malang still faces the problem, it is the low ability in giving the food composition for PE goat. This flaw affects the quality of the produced milk. It needs the artificial science of the milk nutrients contain in order to determine the food composition to produce premium milk with the optimum nutrients contain. The writer uses the method of the Artificial Neural Network (ANN) and the Particle Swarm Optimization (PSO) to make the modeling of goat food in optimizing the content of goat milk nutrients. In the analysis of the examination that is done with the case of 32 kg goat weight, also the food type used is the 70 % Odot grass and 30% Raja grass can increase the nutrients contain of the protein milk for 3.42% and decrease the fat nutrients contain for 5.19%.

Key Words : Goat Milk, Optimization, Artificial Neural Network (ANN), Particle Swarm Optimization (PSO), the food nutrients contain

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xii
DAFTAR GAMBAR xv
DAFTAR PERSAMAAN xvii
DAFTAR LAMPIRAN xix
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 4
1.6 Sistematika Pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 6
2.1 Kajian Pustaka 6
2.2 Jenis Kambing 9
2.3 Bahan Pakan Kambing 10
2.4 Kandungan Nutrisi Pakan Kambing Peranakan Etawa (PE) 11
2.5 Kandungan Gizi Susu Kambing 13
2.6 *Artificial Neural Network (ANN)* 14
2.6.1 Konsep Dasar Artificial Neural Network (ANN) 15
2.6.2 Arsitektur *Artificial Neural Network* (ANN) 15
2.6.3 *Feed Forward Neural Networks* 17
2.6.4 *Backpropagation* 18
2.6.5 Proses Normalisasi 19

2.6.6 Proses Denormalisasi 20
2.6.7 *Mean Square Error* (MSE) 20
2.7 Algoritma *Particle Swarm Optimization* 20
2.7.1 Parameter PSO 22
2.7.2 Penerapan PSO 23
BAB 3 METODOLOGI PENELITIAN 27
3.1 Studi Literatur 27
3.2 Analisa Kebutuhan 28
3.2.1 Deskripsi Umum Sistem 28
3.2.2 Spesifikasi Lingkungan Implementasi 28
3.3 Pengumpulan Data 28
3.4 Perancangan Desain Sistem 29
3.5 Implementasi Sistem 30
3.6 Pengujian dan Analisis 30
3.7 Kesimpulan dan Saran 30
BAB 4 PERANCANGAN 31
4.1 Alir Perancangan Sistem 31
4.1.1 Perhitungan Kandungan Nutrisi Pakan Hijauan Menggunakan ANN 32
4.1.2 Perhitungan Optimasi Komposisi Pakan Menggunakan PSO 40
4.1.3 Proses inisialisasi partikel 42
4.1.4 Proses menghitung fungsi obyektif PSO 43
4.1.5 Proses memperbarui *pBest* 45
4.1.6 Proses mencari *gBest* 46
4.1.7 Proses memperbarui kecepatan partikel 47
4.1.8 Proses memperbarui posisi partikel 48
4.1.9 Proses *Random Injection* 49
4.2 Perhitungan Manualisasi ANN *Backpropagation* 50
4.2.1 Analisis Permasalahan 50
4.2.2 Proses Normalisasi Data Latih 51
4.2.3 Proses Perhitungan Manual *Particle Swarm Optimization* (PSO) 68

4.3 Perancangan Antarmuka 80
4.3.1 Proses Optimasi 80
4.3.2 Halaman Kesimpulan 82
4.4 Perancangan Pengujian 83
4.4.1 Pengujian terhadap maksimal iterasi pada data latih ANN 83
4.4.2 Pengujian terhadap kombinasi nilai bobot inersia minimum dan maksimum 84
4.4.3 Pengujian terhadap nilai konstanta akselerasi 84
4.4.4 Pengujian terhadap nilai konstanta pembatasan kecepatan 85
4.4.5 Pengujian terhadap banyaknya jumlah partikel 86
4.4.6 Pengujian terhadap jumlah maksimum iterasi 86
4.4.7 Pengujian terhadap kombinasi variasi output yang dioptimalkan 87
BAB 5 IMPLEMENTASI 88
5.1 Lingkungan Implementasi 88
5.1.1 Lingkungan Perangkat Keras 88
5.1.2 Lingkungan Perangkat Lunak 88
5.1.3 Batasan Sistem 88
5.2 Implementasi Algoritma 89
5.2.1 Implementasi algoritma *Artificial Neural Network* (ANN) 89
5.2.2 Implementasi algoritma *Particle Swarm Optimization* (PSO) 98
5.3 Implementasi *User Interface* 104
BAB 6 PENGUJIAN DAN PEMBAHASAN 106
6.1 Pengujian 106
6.2 Hasil Pengujian 106
6.2.1 Hasil Pengujian terhadap maksimal iterasi pada data latih ANN 106
6.2.2 Hasil Pengujian Kombinasi Nilai Bobot Inersia 107
6.2.3 Hasil Pengujian terhadap nilai konstanta akselerasi 109
6.2.4 Hasil Pengujian terhadap nilai konstanta pembatasan kecepatan 111
6.2.5 Hasil Pengujian terhadap banyaknya jumlah partikel 113

6.2.6 Hasil Pengujian terhadap jumlah maksimum iterasi 115
6.2.7 Hasil Pengujian terhadap kombinasi variasi *output* yang dioptimalkan 117
BAB 7 PENUTUP 120
7.1 Kesimpulan 120
7.2 Saran 121
DAFTAR PUSTAKA 122
LAMPIRAN 1 SURAT PERMOHONAN DATA SKRIPSI 125
LAMPIRAN 2 Data kandungan gizi susu 126
LAMPIRAN 3 HASIL WAWANCARA PAKAR 132
LAMPIRAN 4 HASIL PENGUJIAN MAKSIMAL ITERASI ANN 136
LAMPIRAN 5 Hasil Pengujian Kombinasi Nilai Inersia 138
LAMPIRAN 6 Hasil Pengujian Konstanta Akselerasi 139
LAMPIRAN 7 Hasil Pengujian Konstanta Pembatasan kecepatan 140
LAMPIRAN 8 Hasil Pengujian Jumlah Partikel 141
LAMPIRAN 9 Hasil Pengujian Maksimal Iterasi PSO 143
LAMPIRAN 10 Hasil Pengujian Variasi Output 144

DAFTAR TABEL

Tabel 2. 1 Kajian Pustaka 7
Tabel 2.2 Kandungan Nutrisi Pakan Kambing 11
Tabel 2.3 Kandungan Gizi Susu Kambing 13
Tabel 2.4 Penggolongan kualitas susu kambing segar berdasarkan karakteristiknya 14
Tabel 3.1 Analisis Kebutuhan Data 29
Tabel 4.1 Kandungan Nutrisi rumput Odot dan raja 51
Tabel 4.2 Total Kandungan Rumput 51
Tabel 4.3 Batas Bawah dan Batas Atas Kandungan Pakan 52
Tabel 4.4 Data Latih Sebelum Normalisasi 52
Tabel 4.5 Data Uji Sebelum Normalisasi 52
Tabel 4.6 Tabel Minimal dan maksimal 53
Tabel 4.7 Data Latih Sesudah Normalisasi 53
Tabel 4.8 Data Uji Sesudah Normalisasi 54
Tabel 4.9 Data ke-1 Perlakuan Premium Sesudah Normalisasi 54
Tabel 4.10 Bobot *Input Layer* ke *Hidden Layer* 55
Tabel 4.11 Jumlah Z_in_j Bobot Sinyal *Input* 55
Tabel 4.12 Hasil Perhitungan Sinyal *Input* dengan Fungsi Sigmoid biner 56
Tabel 4.13 Bobot *Hidden Layer* ke *Output Layer* 56
Tabel 4.14 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer* 57
Tabel 4.15 Hasil Perhitungan Sinyal *Output* dengan Fungsi Sigmoid biner 58
Tabel 4.16 Target Data ke-1 Sesudah Normalisasi 58
Tabel 4.17 Hasil Perhitungan *Error Output* ke Target 59
Tabel 4.18 Hasil Perhitungan Perubahan Bobot *Output Layer* 59
Tabel 4.19 Hasil Menjumlahkan Delta Unit *Hidden* ke *Output* 60
Tabel 4.20 Hasil Menghitung Informasi *Error* 61
Tabel 4.21 Hasil Perubahan Bobot *Hidden Layer* 61
Tabel 4.22 Hasil Hitung *Error* pada Iterasi Pertama 62
Tabel 4.23 Bobot Baru pada Unit *Hidden* 62

Tabel 4.24 Bobot Terakhir pada Unit *Hidden* Iterasi pertama 63
Tabel 4.25 Bobot Baru pada Unit *Output* 63
Tabel 4.26 Hasil *Error* Iterasi ke-2 64
Tabel 4.27 Bobot *Input Layer* ke *Hidden Layer* Data Uji 64
Tabel 4.28 Jumlah Z_in_j Bobot Sinyal *Input* Data Uji 65
Tabel 4.29 Hasil Perhitungan Sinyal *Input* Data Uji 66
Tabel 4.30 Bobot dari *Hidden Layer* ke *Output Layer* Data Uji 66
Tabel 4.31 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer* Data Uji 66
Tabel 4.32 Hasil Perhitungan Sinyal *Output* Data Uji 67
Tabel 4.33 Hasil Denormalisasi Data Uji 68
Tabel 4.34 *Range* Minimal dan Maksimal 68
Tabel 4.35 Inisialisasi partikel PSO 69
Tabel 4.36 Kecepatan Awal Partikel 69
Tabel 4.37 Bobot *input* ke *hidden layer* 70
Tabel 4.38 Jumlah bobot sinyal *input* 70
Tabel 4.39 Hasil Perhitungan sinyal *Input* dengan Aktivasi *Sigmoid Biner* 71
Tabel 4.40 Bobot dari *Hidden Layer* ke *Output Layer* 71
Tabel 4.41 Sinyal *output* 72
Tabel 4.42 Hasil Sinyal *Output* dengan Fungsi Aktivasi *Sigmoid Biner* 73
Tabel 4.43 Hasil Denormalisasi Output 73
Tabel 4.44 Fungsi objektif PSO pada inisialisasi awal partikel 74
Tabel 4.45 pBest untuk inisialisasi partikel 74
Tabel 4.46 gBest untuk inisialisasi partikel 75
Tabel 4.47 Nilai bobot inersia 75
Tabel 4.48 Nilai pembatasan kecepatan maksimum setiap dimensi 76
Tabel 4.49 Kecepatan baru pada iterasi pertama 76
Tabel 4.50 Posisi baru pada iterasi pertama 76
Tabel 4.51 Fungsi objektif PSO pada iterasi pertama 77
Tabel 4.52 pBest pada iterasi pertama 78
Tabel 4.53 *gBest* pada iterasi pertama 78
Tabel 4.54 Posisi baru di iterasi 3 78

Tabel 4.55 Posisi baru setelah *random injection* 79
Tabel 4.56 gBest pada iterasi ke-3 79
Tabel 4.57 Pengujian maksimal iterasi data latih ANN 83
Tabel 4.58 Pengujian kombinasi nilai bobot inersia 84
Tabel 4.59 Pengujian nilai konstanta akselerasi 84
Tabel 4.60 Pengujian nilai konstanta pembatasan kecepatan 85
Tabel 4.61 Pengujian banyaknya jumlah partikel 86
Tabel 4.62 Pengujian jumlah maksimum iterasi 86
Tabel 4.63 Perancangan Pengujian Variasi Output 87
Tabel 6.1 Hasil pengujian maksimal iterasi data latih ANN 106
Tabel 6.2 Hasil Pengujian kombinasi nilai bobot inersia 108
Tabel 6.3 Hasil Pengujian nilai konstanta akselerasi 110
Tabel 6.4 Hasil Pengujian nilai konstanta pembatasan kecepatan 112
Tabel 6.5 Hasil Pengujian banyaknya jumlah partikel 114
Tabel 6.6 Hasil Pengujian jumlah maksimum iterasi 116
Tabel 6.7 Hasil Pengujian Variasi *Output* kualitas Premium 118
Tabel 6.8 Hasil pengujian kenaikan protein dan penurunan lemak 118

DAFTAR GAMBAR

Gambar 2.1 Kambing Potong 10
Gambar 2.2 Kambing Perah 10
Gambar 2.3 Model Neuron 14
Gambar 2.4 Struktur ANN 15
Gambar 2.5 Arsitektur Jaringan Syaraf Tiruan Layer Tunggal 16
Gambar 2.6 Arsitektur Jaringan Syaraf Tiruan Multi Layer 17
Gambar 2.7 Arsitektur Jaringan Syaraf Tiruan Lapisan Kompetitif 17
Gambar 2.8 Dasar prosedur PSO 21
Gambar 3.1 Diagram Alir Metode Penelitian 27
Gambar 3.2 Perancangan Sistem 30
Gambar 4.1 Diagram Alir Perancangan Sistem 31
Gambar 4.2 Diagram Alir Proses ANN 32
Gambar 4.3 Diagram Alir Proses Normalisasi 33
Gambar 4.4 Diagram Alir Proses Pelatihan ANN 34
Gambar 4.5 Diagram Alir Proses *Feedforward* pada ANN 35
Gambar 4.8 Diagram Alir Proses Denormalisasi 39
Gambar 4.9 Diagram alir proses algoritma PSO 41
Gambar 4.10 Diagram alir proses inisialisasi partikel 43
Gambar 4.11 Diagram alir proses menghitung fungsi obyektif 44
Gambar 4.12 Diagram alir proses memperbarui pBest 45
Gambar 4.15 Diagram alir proses memperbarui posisi partikel 48
Gambar 4.16 Diagram alir proses *random injection* 49
Gambar 4.17 Arsitektur Jaringan Kasus Optimasi Kandungan Gizi Susu 50
Gambar 4.18 Rancangan Antar Muka Proses Optimasi 82
Gambar 4.19 Rancangan Antar Muka Hasil kesmpulan 83
Gambar 5.1 *Source code* Proses Normalisasi 90
Gambar 5.2 *Source code* Proses *Feedforward* 93
Gambar 5.3 Algoritma *Backpropagation* 97
Gambar 5.4 Proses denormalisasi 98

Gambar 5.5 Proses Inisialisasi Partikel 99
Gambar 5.6 Proses Hitung Fitness 100
Gambar 5.7 Proses Pencarian PBest 100
Gambar 5.8 Proses Pencarian G_Best 101
Gambar 5.9 Proses Update Kecepatan 102
Gambar 5.10 Proses Update Posisi 103
Gambar 5.11 Proses Random Injection 103
Gambar 5.12 Implementasi Halaman Optimasi 104
Gambar 5.13 Implementasi halaman hasil Kesimpulan 105
Gambar 6.1 Grafik pengujian iterasi maskimum data latih ANN 107
Gambar 6.2 Grafik pengujian Kombinasi Nilai inersia 109
Gambar 6.3 Grafik pengujian nilai Konstanta Kecepatan 111
Gambar 6.4 Grafik Konstanta Pembatasan Kecepatan 113
Gambar 6.5 Grafik Pengujian Jumlah Partikel 114
Gambar 6.6 Grafik waktu komputasi jumlah partikel 115
Gambar 6.7 Grafik Pengujian maksimum iterasi PSO 116
Gambar 6.8 Hasil variasi *output* Protein Optimal dan Lemak Minimal 118
Gambar 6.9 Hasil Kenaikan Protein dan Penurunan Lemak 119

DAFTAR PERSAMAAN

Persamaan 2.1 11
Persamaan 2.2 11
Persamaan 2.3 18
Persamaan 2.4 18
Persamaan 2. 5 18
Persamaan 2.6 18
Persamaan 2. 7 18
Persamaan 2.8 18
Persamaan 2.9 19
Persamaan 2.10 19
Persamaan 2.11 19
Persamaan 2.12 19
Persamaan 2.13 19
Persamaan 2.14 19
Persamaan 2.15 19
Persamaan 2.16 20
Persamaan 2. 17 20
Persamaan 2. 18 20
Persamaan 2.19 24
Persamaan 2.20 24
Persamaan 2.21 24
Persamaan 2.22 24
Persamaan 2.23 24
Persamaan 2.24 25
Persamaan 2.25 25
Persamaan 2.26 25
Persamaan 2.27 25
Persamaan 2.28 26
Persamaan 2.29 26

Persamaan 2.30……. 26

DAFTAR LAMPIRAN

Lampiran 1 Surat Permohonan data Skripsi.....125
Lampiran 2 Data Kandungan Gizi Susu.....126
Lampiran 3 Hasil Wawancara Pakar 132
Lampiran 4 Hasil Pengujian Maksimal Iterasi ANN 135
Lampiran 5 Hasil Pengujian Kombinasi Nilai Inersia.....137
Lampiran 6 Hasil Pengujian Konstanta Akselerasi 138
Lampiran 7 Hasil Pengujian Konstanta Pembatasan kecepatan.....139
Lampiran 8 Hasil Pengujian Jumlah Partikel..... 140
Lampiran 9 Hasil Pengujian Maksimal Iterasi PSO.....142
Lampiran 10 Hasil Pengujian Variasi Output.....143

BAB 1 PENDAHULUAN

1.1 Latar belakang

Susu merupakan bahan makanan yang dapat dikatakan sebagai makanan yang sempurna karena pada susu terdapat kandungan gizi yang sangat diperlukan oleh tubuh, seperti protein, karbohidrat, lemak dan vitamin. Selain itu, susu juga satu dari beberapa sumber protein hewani yang mudah diperoleh serta dapat dimanfaatkan untuk memenuhi kebutuhan pangan yang bergizi di masyarakat (Mulyanto, 2006). Tingkat kesadaran masyarakat yang semakin tinggi mengenai pentingnya susu mengakibatkan adanya peningkatan dalam hal konsumsi susu pada masyarakat, namun susu yang dikonsumsi masyarakat masih didominasi oleh susu impor (Marini Putri Marwah, 2010). Impor susu disebabkan karena produksi susu segar yang dihasilkan dari sapi perah yang merupakan penghasil utama susu segar di Indonesia masih belum dapat memenuhi permintaan susu masyarakat yang semakin tinggi disebabkan oleh skala usaha 2-3 ekor sapi perah per peternak (Nugroho, 2010). Salah satu alternatif untuk memenuhi kebutuhan produksi susu segar, yaitu susu kambing (Marini Putri Marwah, 2010).

Susu kambing dinilai lebih berkualitas dari segi kandungan gizinya dibandingkan dengan susu sapi. Karakteristik susu kambing dibandingkan susu sapi adalah warnanya lebih putih. Hal ini dikarenakan kandungan vitamin A pada susu kambing tidak tersusun sebagai pigmen karotenoid seperti susu sapi. Selain itu globula lemak susunya lebih kecil sehingga lemak susu kambing lebih mudah dicerna, mengandung mineral kalsium, fosfor, vitamin A, E, dan B kompleks yang tinggi dan dapat diminum oleh orang yang alergi terhadap susu sapi, atau untuk orang-orang yang mengalami berbagai gangguan pencernaan (Yayu Zuriati, 2011). Susu kambing memiliki ukuran rata-rata butiran lemak sebesar 2 mikrometer, lebih kecil dari pada ukuran butiran lemak susu sapi yang mencapai 2,5-3,5 mikrometer. Ukuran butiran lemak yang lebih kecil ini membuat lemak susu kambing lebih tersebar dan homogen sehingga lebih mudah dicerna oleh sistem pencernaan manusia (Purbayanto, 2009). Protein susu kambing lebih mudah larut dan lebih mudah diserap serta lebih rendah dalam memicu alergi oleh tubuh sehingga mengindikasikan bahwa kualitas protein susu kambing lebih baik dibandingkan dengan susu sapi (Aliaga, 2003).

Tingginya produksi susu dan kualitas kandungan gizi susu sangat dipengaruhi oleh beberapa faktor salah satunya, yaitu faktor pakan (Prihatminingsih, 2015). Pakan berfungsi sebagai pembangunan dan pemeliharaan tubuh, sumber energi, produksi dan pengatur proses-proses yang ada di dalam tubuh (Subekti, 2009). Bagian peternakan kambing peranakan etawa (PE) di Unit Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang masih menghadapi permasalahan, yaitu rendahnya kemampuan dalam memberikan *treatment* komposisi pakan terhadap kambing peranakan etawa. Kekurangan tersebut selain mengakibatkan lambatnya pertumbuhan produksi susu juga berpengaruh terhadap kualitas susu yang dihasilkan (Nugroho, 2010). Susu

kambing yang berkualitas merupakan susu dengan kandungan gizi yang sesuai Standar Nasional Indonesia nomor 01-3141-1998 (Yayu Zuriati, 2011). Pihak pakan UPT tidak pernah memberikan komposisi pakan yang berbeda-beda karena kurangnya pengetahuan dalam menentukan komposisi pakan untuk mendapatkan kandungan gizi susu yang optimal. Dalam hal ini diperlukan pengetahuan rekayasa kandungan gizi susu untuk menentukan komposisi pakan yang akan diberikan untuk mendapatkan kandungan gizi dengan kualitas yang melebihi standar. Susu premium dengan kandungan gizi yang optimal berpotensi dalam memenuhi kebutuhan gizi pada masyarakat, serta dapat menyembuhkan beberapa penyakit seperti asma dan TBC (Atmiyati, 2001).

Permasalahan rekayasa kandungan gizi susu kambing agar dapat optimal tersebut dapat diselesaikan dengan beberapa metode pemodelan. Perkembangan terakhir pemodelan prediksi dengan menggunakan *Artificial Neural Network* (ANN) merupakan salah satu pemodelan yang banyak digunakan dalam prediksi ekonomi, keuangan, bisnis dan industry (Raharjo, 2013). Kemampuan ANN untuk pelatihan data *training* dilakukan dengan menggunakan *backpropagation*, namun lambatnya laju konvergensi dan terjebak dalam minimum local merupakan beberapa kelemahan dari metode *backpropagation* (Raharjo, 2013). Untuk mengatasi kelemahan tersebut, dapat digunakan beberapa pendekatan optimasi seperti *genetic algoritma* (GA), *Particle Swarm Optimization* (PSO) dan *ant colony optimization* (ACO) (S.-C. Chu, 2004).

Penelitian terdahulu memberikan kesimpulan bahwa dibanding ACO dan GA, secara individu dengan PSO memiliki kinerja terbaik karena dapat lebih cepat mendapatkan hasil yang optimal. (Raharjo, 2013). Kemudian dikembangkan model dengan menggunakan dua metode yaitu gabungan PSO-ANN untuk menguji beberapa data klasifikasi yang menghasilkan rata-rata yang cukup efektif dalam mencapai konvergensi dan ruang global minimum. Dalam penelitian sebelumnya yang membahas prediksi laju inflasi menggunakan ANN dan PSO oleh Joko S. Dwi Raharjo (2013) telah berhasil memberikan kesimpulan bahwa penerapan model aw(*atribut weight)* PSO-ANN dapat meningkatkan tingkat akurasi dalam melakukan prediksi laju inflasi berdasarkan atribut laju inflasi dan indeks harga konsumen (IHK) di 4 kota besar di Indonesia yaitu Jakarta, Surabaya, Bandung dan Medan. Penelitian lain yang membahas optimasi parameter dalam pembuatan genteng pada sebuah pabrik yang menggunakan metode ANN dan PSO oleh Thitipong Navalertporn dan Nitin V. Afzulpurkar (2011) juga membuktikan bahwa dengan menggunakan kombinasi dua metode ini menghasilkan parameter-parameter yang optimal sehingga meningkatkan hasil produksi dari 60% menjadi 97% serta mendapatkan hasil yang lebih akurat daripada hanya menggunakan metode *backpropagation* saja.

Berdasarkan latar belakang yang telah diuraikan di atas, penulis berinisiatif untuk merancang sistem berbasis komputer untuk memberikan solusi optimasi kandungan gizi susu kambing dalam penelitian tugas akhir yang berjudul " Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) untuk Optimasi Kandungan Gizi Susu Menggunakan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO)". Penulis berharap hasil penelitian ini dapat membantu

peternak kambing peranakan etawa dalam mengoptimalkan kandungan gizi susu yang dihasilkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, penulis merumuskan beberapa masalah yaitu sebagai berikut:

1. Bagaimana mengimplementasikan metode *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO) untuk menyelesaikan permasalahan Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) untuk Mengoptimalkan Kandungan Gizi Susu Kambing?
2. Berapa nilai parameter optimal untuk optimasi dari kandungan gizi susu kambing yang menggunakan metode Artificial Neural Network (ANN) dan *Particle Swarm Optimization* (PSO)?
3. Berapa hasil kualitas solusi untuk optimasi dari kandungan gizi susu kambing yang menggunakan metode Artificial Neural Network (ANN) dan *Particle Swarm Optimization* (PSO)?

1.3 Tujuan

1. Menerapkan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO) untuk menyelesaikan permasalahan Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) untuk Mengoptimalkan Kandungan Gizi Susu Kambing.
2. Mengetahui nilai parameter yang optimal untuk optimasi dari kandungan gizi susu kambing yang menggunakan metode Artificial Neural Network (ANN) dan *Particle Swarm Optimization* (PSO).
3. Mengetahui hasil kualitas solusi untuk optimasi dari kandungan gizi susu kambing yang menggunakan metode Artificial Neural Network (ANN) dan *Particle Swarm Optimization* (PSO).

1.4 Manfaat

Berdasarkan rumusan masalah, penelitian ini diharapkan dapat bermanfaat untuk berbagai pihak. Manfaat dari penelitian ini adalah sebagai berikut:

1. Membantu para peternak kambing peranakan etawa (PE) dalam membuat kombinasi pakan kambing PE yang terbaik.
2. Membantu para peternak kambing peranakan etawa (PE) untuk meningkatkan kandungan gizi susu kambing dengan tetap memperhatikan nutrisi yang dibutuhkan oleh kambing PE.

1.5 Batasan masalah

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

1. Jenis kambing yang diteliti adalah kambing perah, yaitu kambing peranakan etawa (PE) betina pada masa laktasi dengan bobot badan 32 kg, 34 kg, 36 kg, 38 kg dan 40 kg.
2. Jenis pakan yang digunakan adalah 10 jenis bahan pakan hijauan yaitu rumput taiwan, rumput odot, rumput setaria, rumput raja, rumput B.Decumbens, kaliandra, gamal, lamtoro, turi dan tarum.
3. Jenis pakan yang dapat dimodelkan hanya dua jenis.
4. Terdapat 4 output kandungan gizi susu yang diteliti, yaitu protein, lemak (*fat*), laktosa dan *density* (berat jenis).
5. Kandungan gizi susu yang dapat dioptimalkan adalah protein, laktosa dan *density* (berat jenis).
6. Kandungan gizi susu yang dapat diminimalkan adalah lemak (*fat*).
7. Kandungan nutrisi dari pakan hijauan berupa Abu, PK, LK dan SK.

1.6 Sistematika Pembahasan

Sistematika pembahasan memberikan Gambaran dan deskripsi singkat mengenai isi dari tugas akhir penyusun yang meliputi beberapa bab sebagai berikut:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika pembahasan mengenai tugas akhir Pemodelan Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) menggunakan *Artificial Neural Network* dan *Particle Swarm Optimization*.

BAB II Landasan Kepustakaan

Menjelaskan tentang dasar teori dan referensi yang mendasari pembuatan sistem Pemodelan Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) menggunakan *Artificial Neural Network dan Particle Swarm Optimization*.

BAB III Metode Penelitian

Menjelaskan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, metode pengambilan data, analisi kebutuhan, perancangan sistem, implementasi pengujian dan analisis serta pengambilan kesimpulan.

BAB IV Perancangan

Menjelaskan tentang perancangan database, perancangan antarmuka dan perancangan uji coba dalam pembuatan perangkat lunak sistem pemodelan optimasi kandungan gizi susu kambing yang akan dibuat.

BAB V Implementasi

Membahas implementasi dari sistem Pemodelan Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) menggunakan *Artificial Neural Network* dan *Particle Swarm Optimization* sesuai dengan perancangan sistem yang telah dibuat serta implementasi antarmuka.

BAB VI Pengujian dan Analisis

Menjelaskan proses dan hasil dari pengujian terhadap sistem yang telah dibangun serta analisis untuk memastikan bahwa program telah sesuai dengan perancangan.

BAB VII Penutup

Bab ini terdiri dari kesimpulan yang diperoleh dari pembuatan dan perancangan perangkat lunak yang dibuat dalam skripsi ini serta saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan untuk menunjang skripsi pembuatan aplikasi Pemodelan Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) menggunakan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO) berisi uraian dan pembahasan tentang penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu.

2.1 Kajian Pustaka

Pada penelitian sebelumnya, belum ada jurnal yang membahas tentang optimasi pakan kambing perah, sehingga pada kajian pustaka, peneliti hanya menggunakan referensi jurnal yang membahas tentang metode ANN-PSO. Pada referensi jurnal yang pertama dengan judul "Model *Artificial Neural Network* Berbasis *Particle Swarm Optimization* untuk Prediksi Laju Inflasi" oleh Joko S. Dwi Raharjo membahas tentang perencanaan dalam membuat strategi ekonomi untuk mengantisipasi perkembangan laju inflasi dan perencanaan keuangan ke depan. Kondisi naik turunnya laju inflasi ini akan mengakibatkan kondisi pertumbuhan ekonomi pada suatu Negara, sehingga prediksi laju inflasi sangat dibutuhkan untuk membuat strategi agar pertumbuhan ekonomi tidak menurun. Dengan penelitian tersebut yang menerapkan model PSO-ANN meningkatkan akurasi dalam prediksi laju inflasi berdasarkan atribut laju inflasi dan indeks harga konsumen (IHK) empat kota besar yaitu Jakarta, Surabaya, Bandung dan Medan.

Penelitian kedua dengan judul "Model Prediksi Kebangkrutan Berbasis *Neural Network* dan *Particle Swarm Optimization*" oleh Isye Arieshanti, Yudhi Purwananto membahas model klasifikasi untuk memprediksi kebangkrutan suatu perusahaan. Model dikembangkan berdasarkan metode yang berbasis ANN dengan metode PSO. Metode-metode yang berbasis ANN bertugas sebagai penentu klasifikasi sedangkan PSO bertugas sebagai pemilih fitur dan penentu parameter-parameter (*learning rate* dan *epoch*) optimal model. Dari hasil uji coba dapat disimpulkan bahwa model yang menggabungkan ANN dengan PSO terbukti memiliki performa yang cukup baik, yaitu sekitar 72%-75%.

Pada penelitian ketiga dengan judul "Optimasi Jaringan Syaraf Tiruan *backpropagation* Menggunakan *Particle Swarm Optimization* Untuk Deteksi Penderita Penyakit Jantung" oleh Hargianti Henni Oktawandari membahas tentang klasifikasi apakah seseorang terindikasi penyakit jantung, dalam penelitian kali ini diterapkan *Artificial Neural Network* atau Jaringan Syaraf Tiruan model *backpropagation* dan Jaringan Syaraf Tiruan model *backpropagation* yang bobotnya dioptimasi menggunakan *Particle Swarm Optimization* dengan tujuan meminimalkan *mean squared error* (MSE). Data yang digunakan berasal dari UCI *Machine Learning Respository* dengan 270 hasil observasi partisipan terdiri dari 13 variabel yang ditentukan untuk melakukan deteksi. Hasil pengujian menunjukkan

pada percobaan kedua nilai akurasi BPNN+PSO lebih tinggi dari BPNN saja, yakni 87.7% dan nilai RMSE menurun dari 1.0862 menjadi 0.3403.

Penelitian keempat dengan judul "*A new personalized ECG signal classification algorithm using Block-based Neural Network and Particle Swarm Optimization*" oleh Shirin Shadmand dan Behbood Mashoufi membahas tentang klasifikasi detak jantung pasien melalui sinyal EGC dengan menggunakan BBNN yang dilatih melalui metode PSO menampilkan sistem yang hasilnya memiliki akurasi tinggi yaitu 97% untuk klasifikasi sinyal EGC. Kecepatan konvergensi sangat tinggi karena menggunakan algoritma PSO. Sedangkan BBNN digunakan untuk klasifikasi yang sangat mampu menangani perubahan parameter dari sinyal EGC.

Penelitian kelima dengan judul "*Optimization of tile manufacturing process using Particle Swarm Optimization*" oleh Thitipong Navalertporn dan Nitin V. Afzulpurkar (2011) membahas tentang bagaimana cara mendapatkan parameter-parameter yang optimal dalam proses pembuatan genteng, sebab selama ini parameter yang didapatkan berdasarkan dari pengalaman, ketrampilan dan intuisi para pekerja dalam pabrik serta upaya untuk mengoptimalkan parameter dilakukan melalui *trial* dan *error* sehingga tidak ada jaminan bahwa parameter tersebut merupakan parameter yang optimal. Karena permasalahan tersebut, maka dilakukan sebuah penelitian optimasi parameter dengan menggunakan ANN dan PSO. Hasilnya membuktikan bahwa dengan menggunakan kombinasi dua metode ini menghasilkan parameter-parameter yang optimal sehingga meningkatkan hasil produksi dari 60% menjadi 97% serta mendapatkan hasil yang lebih akurat daripada hanya menggunakan metode *backpropagation* saja.

Metode *hybrid Artificial Neural Network* (ANN) dengan *Particle Swarm Optimization* (PSO) merupakan metode yang saling melengkapi, di dalam metode ANN selain memiliki kemampuan dalam meningkatkan akurasi, ANN juga memiliki kelemahan yaitu konvergensi yang lambat, sedangkan metode PSO dapat mempercepat laju konvergensi. Oleh karena itu, penulis menggunakan gabungan metode ANN dan PSO dalam memodelkan pakan kambing peranakan etawa untuk optimasi kandungan gizi susu kambing.

Untuk lebih jelasnya dalam memahami terhadap semua yang telah dipaparkan sebelumnya, maka penulis meringkasnya ke dalam bentuk Tabel Kajian Pustaka yang ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Kajian Pustaka

No.	Judul	Object	Metode	Hasil
1	Model *Artificial Neural Network* Berbasis	Object: Laju Inflasi	Metode: *Artificial Neural Network* dan *Particle Swarm Optimization*	Penerapan model PSO-ANN meningkatkan akurasi dalam prediksi laju inflasi berdasarkan

	Particle Swarm Optimization untuk Prediksi Laju Inflasi			atribut laju inflasi dan indeks harga konsumen (IHK) empat kota besar yaitu Jakarta, Surabaya, Bandung dan Medan.
2	Model Prediksi Kebangkrutan Berbasis *Neural Network* dan *Particle Swarm Optimization*	Object : Prediksi Kebangkrutan	Metode: *Artificial Neural Network* dan *Particle Swarm Optimization*	Semua model yang menggunakan gabungan metode ANN yang dioptimasi dengan menggunakan PSO, memiliki nilai akurasi yang lebih tinggi jika dibandingkan dengan nilai akurasi model yang tidak menggunakan seleksi fitur
3	Optimasi Jaringan Syaraf Tiruan *Backpropagation* Menggunakan *Particle Swarm Optimization* Untuk Deteksi Penderita Penyakit Jantung	Object : 13 fitur yang digunakan untuk prediksi peyakit jantung	Metode: *Backpropagation* dan *Particle Swarm Optimization*	Berdasarkan penelitian yang telah dilakukan dapat ditarik kesimpulan bahwa hasil pelatihan menggunakan metode BPNN yang bobotnya dioptimasi dengan PSO nilai akurasinya mencapai 87.7%, lebih tinggi dibanding hasil pelatihan yang hanya menggunakan metode BPNN yaitu sebesar 82.7%. Hasil iterasi yang lebih baik juga ditunjukkan dengan nilai RMSE yang lebih kecil sebesar 0.3403 dibandingkan dengan tanpa menggunakan PSO

4	*A new personalized ECG signal classification algorithm using Block-based Neural Network and Particle Swarm Optimization*	Object : Sinyal *Electrocardiogram* (ECG)	Metode : *Block-based Neural Network* dan *Particle Swarm Optimization*	Dengan menggunakan BBNN yang dilatih melalui metode PSO menampilkan sistem yang memiliki akurasi tinggi yaitu 97% untuk klasifikasi sinyal ECG. Kecepatan konvergensi sangat tinggi karena menggunakan algoritma PSO. Sedangkan BBNN digunakan untuk klasifikasi yang sangat mampu menangani perubahan parameter dari sinyal EGC
5	*Optimization of tile manufacturing process using Particle Swarm Optimization*	Object : Parameter yang dibutuhkan dalam pembuatan genteng	Metode: *Artificial Neural Network* dan *Particle Swarm Optimization*	Hasilnya membuktikan bahwa dengan menggunakan kombinasi dua metode ini menghasilkan parameter-parameter yang optimal sehingga meningkatkan hasil produksi dari 60% menjadi 97% serta mendapatkan hasil yang lebih akurat daripada hanya menggunakan metode *backpropagation* saja.

2.2 Jenis Kambing

Kambing merupakan hewan ternak yang banyak memberikan manfaat seperti daging dan susunya. Jenis kambing dapat dibedakan menjadi kambing potong dan kambing perah (Gunawan, 2013).

a. Kambing Potong

Kambing potong adalah kambing yang diternak untuk produksi daging. Kambing potong memiliki ciri-ciri sedikit lemak dan bulu dan pada umumnya ukuran badannya lebih besar yang ditunjukkan pada Gambar 2.1.

Gambar 2.1 Kambing Potong

b. Kambing Perah

Kambing perah adalah kambing yang diternak untuk produksi susu. Untuk meningkatkan produksi susu kambing, maka dipilih kambing unggul dalam menghasilkan susu. Kambing perah memiliki badan yang lebih kurus dibandingkan dengan kambing potong, yang ditunjukkan pada Gambar 2.2.

Gambar 2.2 Kambing Perah

2.3 Bahan Pakan Kambing

Bahan pakan adalah setiap bahan yang dapat dimakan, disukai, dapat dicerna sebagian atau seluruhnya, dapat diabsorpsi dan bermanfaat bagi ternak. Sedangkan, pakan adalah bahan yang dapat dimakan, dicerna dan diserap baik secara keseluruhan atau sebagian dan tidak menimbulkan keracunan atau tidak mengganggu kesehatan ternak yang mengkonsumsinya. Pakan berfungsi sebagai pembangunan dan pemeliharaan tubuh, sumber energi, produksi, dan pengatur proses-proses dalam tubuh (Subekti, 2009). Pakan merupakan faktor terbesar

untuk menghasilkan produktivitas tinggi, yaitu produksi air susu, pertumbuhan dan produksi kambing (Hidayati, 2012). Kambing adalah hewan ternak ruminansia kecil yang pakan utamanya adalah hijauan. Ternak ruminansia akan berproduksi dengan baik jika tersedia pakan hijauan yang berkualitas secara cukup dan berkesinambungan. Pakan hijauan merupakan kombinasi rumput dan *legume* yang dibutuhkan untuk saling melengkapi unsur nutrisi yang diperlukan oleh ternak (Koten, 2014).

Selain diberi pakan hijauan, kambing juga diberi pakan berupa konsentrat, konsentrat merupakan makanan penguat yang kandungan gizinya tinggi. Meskipun konsentrat memiliki gizi yang baik, namun konsetrat tidak bisa dijadikan makanan tunggal karena dapat menyebabkan gangguan pencernaan pada kambing yang mengakibatkan hewan ternak mati. Sehingga pakan kambing yang baik adalah perpaduan antara hijauan dan konsentrat (Susilawati, 2011). Bahan pakan untuk domba dan kambing pada umumnya di golongkan dalam 4 golongan sebagai berikut (Cahyono, 2006) :

1. Golongan rumput-rumputan
    a. Rumput gajah
    b. Rumput setaria
    c. Rumput benggala
    d. Rumput brachiaria
    e. Rumput raja
    f. Rumput meksiko
    g. Rumput alam
2. Golongan kacang-kacangan atau *leguminose*
    a. Daun lamtoro
    b. Daun turi
    c. Daun gamal
    d. Kaliandra
    e. Albesia
    f. Gliricidia
    g. Siratro
    h. Daun kacang tanah
    i. Daun kacang-kacangan
    j. Daun kacang

3. Hasil limbah pertanian
    a. Daun nangka
    b. Daun waru
    c. Daun dadap
    d. Daun kembang sepatu
    e. Daun pisang
    f. Daun jagung
    g. Daun ketela pohon
    h. Daun ketela rambat
    i. Daun beringin
4. Golongan makanan penguat (konsentrat)
    a. Dedak
    b. Jagung kering
    c. Garam dapur
    d. Bungkil kelapa
    e. Tepung ikan
    f. Bungkil kedelai
    g. Ampas tahu
    h. Ampas kecap

Dalam pemberian pakan, pihak UPT menggunakan persamaan seperti persamaan 2.1.

Banyaknya Pakan(minimal) = $10\% * Bobot\ Kambing$ **2.1**

Banyaknya Pakan(maksimal) = $12\% * Bobot\ Kambing$ **2.2**

2.4 Kandungan Nutrisi Pakan Kambing Peranakan Etawa (PE)

Pakan ternak memegang peranan yang sangat penting dalam usaha peternakan. Pakan ternak harus terjaga kualitas dan kuantitas agar proses perkembangan, produksi dihasilkan menjadi baik. Berikut adalah daftar tabel kandungan dari berbagai bahan pakan hijuan dalam 1 kg yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 Kandungan Nutrisi Pakan Kambing

No.	Bahan Pakan	Abu (%)	PK(%)	LK(%)	SK(%)
1	Rumput Taiwan	18,78	18,78	26,45	1,20

2	Rumput Odot	16,97	17,03	24,84	1,76
3	Rumput Setaria	11,5	8,5	32,5	1,76
4	Rumput Raja	18,6	13,5	24,84	3,5
5	Rumput Brachiaria decumbens	10,6	8,3	38,3	1,2
6	Kaliandra	7,27	21,55	13,96	2,99
7	Gamal	9,60	19,96	17,59	3,85
8	Lamtoro	-	24,2	21,5	3,7
9	Turi	-	29,2	17,2	3,4
10	Tarum	6,42	24,17	-	-
11	Pollar	-	16,5	10	4
12	Bungkil Sawit	-	18,5	12	2,5
13	Empok Jagung	-	8,40	9,72	6,91
14	Bungkil Kedelai	-	43,7	6	0,9
15	Mineral	-	-	-	-

Sumber: disnak.jatimprov

Keterangan:

1. Abu

Abu merupakan zat pakan anorganik yang mengandung unsur-unsur yang dibutuhkan oleh hewan ternak seperti Ca, K, Na, Mg, Fe, P dan CL.

2. Protein kasar (PK)

Protein adalah bahan organik kompleks yang mempunyai berat molekul tinggi dengan peranan yang banyak dan berbeda-beda dalam tubuh. Protein sangat penting untuk efisiensi penggunaan energi dan pertumbuhan otot. Yang diperlihatkan dalam tabel kandungan nutrisi bahan ransum pakan kambing PE adalah protein kasar.

3. Lemak kasar (LK)

Lemak merupakan salah satu kelompok yang termasuk golongan lipida. Lemak terdapat pada hampir semua bahan pangan dengan kandungan yang berbeda-beda. Kandungan lemak dalam pangan adalah lemak kasar yang merupakan kandungan total lipida dalam jumlah yang sebenarnya.

4. Serat kasar (SK)

Serat kasar merupakan sisa-sisa sel tumbuhan yang tahan terhadap reaksi hidrolisis enzim-enzim yang ada pada saluran pencernaan. Komponen utama

penyusun serat kasar berupa karbohidrat. Jumlah serat kasar pada pakan didasarkan pada *feed intake* (jumlah pakan yang dikonsumsi) oleh hewan ternak

2.5 Kandungan Gizi Susu Kambing

Susu segar menurut DEWAN STANDARISASI NASIONAL (1998) dalam Standar Nasional Indonesia nomor 01-3141-1998 didefinisikan sebagai cairan yang berasal dari kambing ternak sehat yang diperah dengan cara pemerahan yang benar, tidak mengalami penambahan atau pengurangan suatu komponen apapun kecuali proses pendinginan dan tanpa mempengaruhi kemurniannya. Analisis karakteristik kualitas susu segar yang dilakukan meliputi berat jenis, kadar protein, kadar lemak, kadar bahan kering dan bahan kering tanpa lemak. Kandungan gizi susu kambing ditunjukkan dalam Tabel 2.3.

Tabel 2.3 Kandungan Gizi Susu Kambing

No	Kandungan	Susu Kambing
1	Protein (g)	3,6
2	Lemak (g)	4,2
3	Karbohidrat (g)	4,5
4	Kalori (g)	69
5	Fosfor I (g)	111
6	Kalsium (g)	134
7	Magnesium (g)	14
8	Besi (g)	0,05
9	Natrium (g)	50
10	Kalium (g)	204
11	Vitamin A (IU)	185
12	Thiamin (mg)	0,05
13	Riboflavin (g)	0,14
14	Niacin (mg)	0,28
15	Vitamin B6 (mg)	0,05

Sumber: Balai Penelitian Veteriner, Bogor (2008)

Susu kambing mempunyai keistimewaan lain yaitu kaya dengan protein, enzim, vitamin A. Protein susu kambing merupakan protein terbaik setelah ASI (Air Susu Ibu) (Mulyanto, 2006). Protein susu kambing lebih mudah larut dan lebih mudah diserap serta lebih rendah dalam memicu alergi oleh tubuh sehingga

mengindikasikan bahwa kualitas protein susu kambing lebih baik dibandingkan dengan susu sapi (Aliaga, 2003). Lemak susu kambing lebih kecil dari susu sapi sehingga lemak susu kambing lebih mudah dicerna, dan dapat diminum oleh orang yang alergi terhadap susu sapi, atau untuk orang-orang yang mengalami berbagai gangguan pencernaan (Blakely, 1991). Berikut ini merupakan tabel kualitas susu kambing ditunjukkan pada Tabel 2.4.

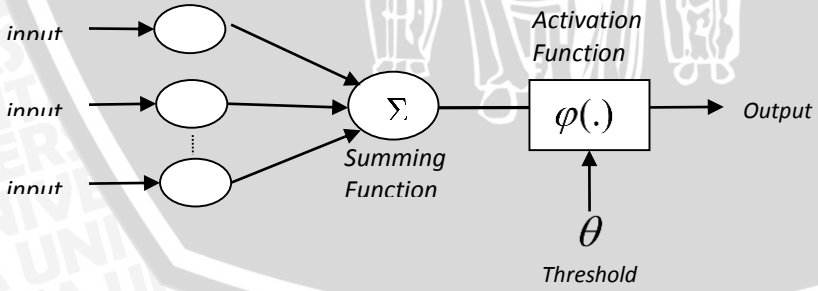
Tabel 2.4 Penggolongan kualitas susu kambing segar berdasarkan karakteristiknya

Karakteristik	Kriteria Berkualitas		
	Premium	Baik	Standar
Protein (%)	>3,70	>3,40-3,70	3,10-3,40
Bahan Kering (%)	>13,00	>12,00-13,00	11,70-12,00

Sumber: Thai Agricultural Standard (2008)

2.6 *Artificial Neural Network (ANN)*

Artificial Neural Network (ANN) atau Jaringan Saraf Tiruan merupakan model penalaran yang didasarkan pada otak manusia. ANN terdiri dari sejumlah prosesor sangat sederhana dan saling berhubungan yang disebut neuron. Neuron yang terhubung dengan pembobotan (*weight*) melewati sinyal dari neuron satu ke neuron yang lain. ANN mampu mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya dan mampu bekerja meskipun beberapa neuronnya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak maka neuron lain dapat dilatih untuk menggantikan fungsi neuron yang rusak tersebut. Model neuron ANN terdiri; fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*) seperti pada Gambar 2.3.



Gambar 2.3 Model Neuron

Cara kerja ANN adalah informasi yang masuk (*input*) akan dikirim ke neuron dengan bobot tertentu dan selanjutnya diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan akan dibandingkan dengan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron.

Jika *input* melewati suatu nilai ambang tertentu, maka neuron akan diaktifkan, jika tidak, maka neuron tidak akan diaktifkan. Neuron yang diaktifkan akan mengirimkan *output* melalui bobot-bobot *output*-nya ke semua neuron yang berhubungan dengannya.

2.6.1 Konsep Dasar Artificial Neural Network (ANN)

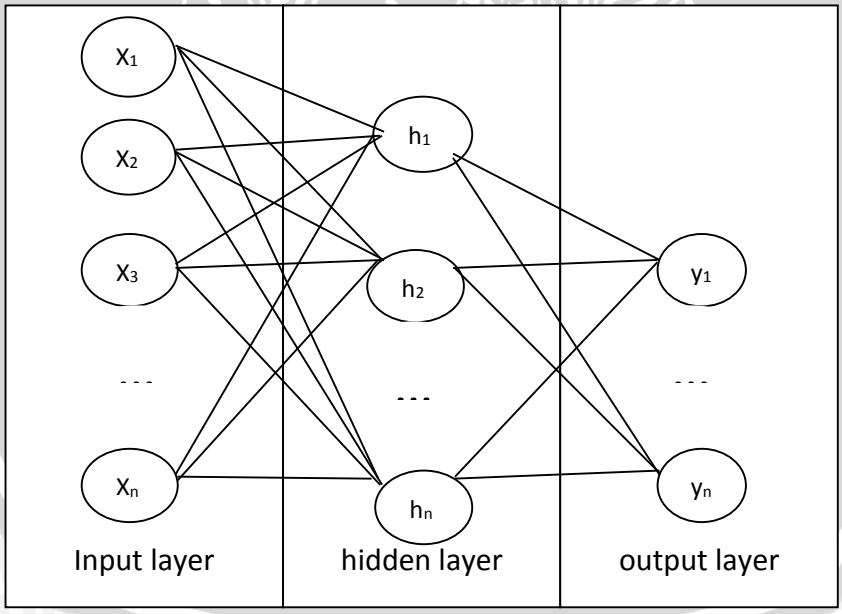
Setiap pola-pola informasi input dan output yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu (Matondang, 2013) :

1. Lapisan Input, unit-unit di dalam lapisan input disebut unit-unit input. Input tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.

2. Lapisan Tersembunyi, unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di mana outputnya tidak dapat secara langsung diamati.

3. Lapisan Output, unit-unit di dalam lapisan output disebut unit-unit output. Output dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

Struktur tersebut digambarkan pada Gambar 2.4.



Gambar 2.4 Struktur ANN

2.6.2 Arsitektur *Artificial Neural Network* (ANN)

JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Matondang, 2013) :

1. Jaringan Layar Tunggal (*Single Layer Network*)

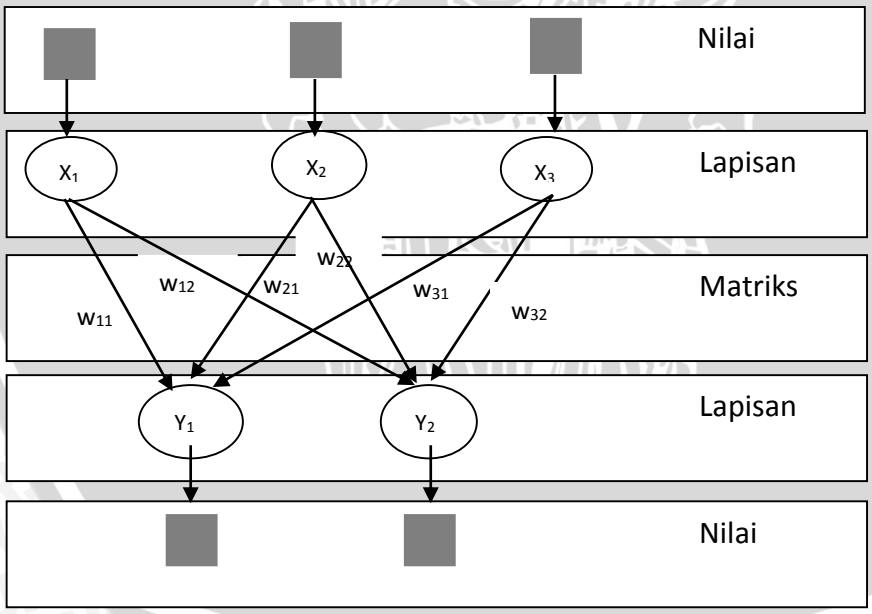
Jaringan dengan lapisan tunggal terdiri dari 1 layer *input* dan 1 layer *output*. Setiap neuron/unit yang terdapat di dalam lapisan/*layer input* selalu terhubung dengan setiap neuron yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Contoh algoritma JST yang menggunakan metode ini yaitu : ADALINE, Hopfield, Perceptron. Ilustrasi jaringan layar tunggal dapat dilihat pada Gambar 2.5.

2. Jaringan Banyak Lapisan (*Multilayer Net*)

Jaringan dengan lapisan jamak ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma Jaringan Syaraf Tiruan yang menggunakan metode ini yaitu: MADALINE, *Backpropagation*, *Neocognitron*. Jaringan banyak lapisan digambarkan seperti Gambar 2.6.

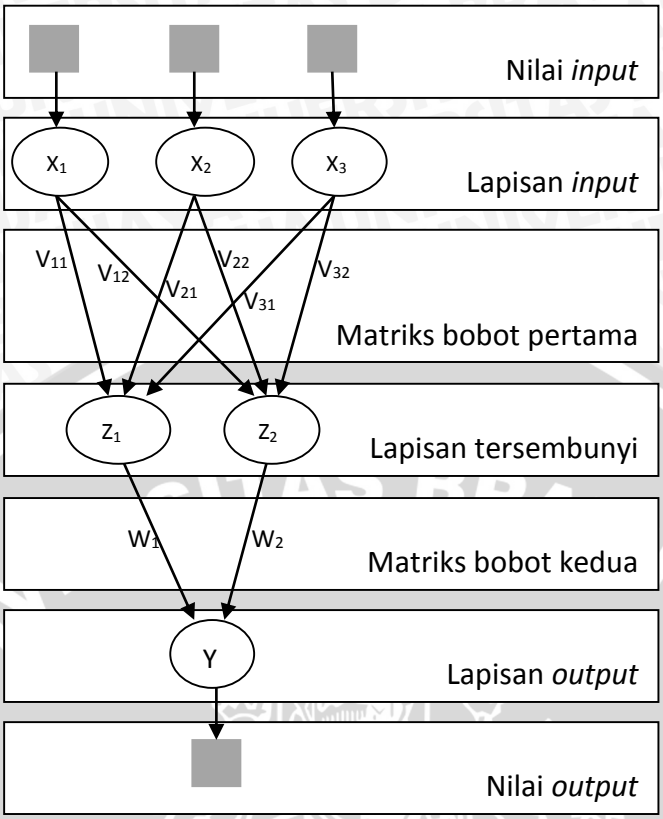
3. Jaringan Lapisan Kompetitif (*Competitive Layer*)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah LVQ. Ilustrasi visual jaringan lapisan kompetitif seperti pada Gambar 2.7.

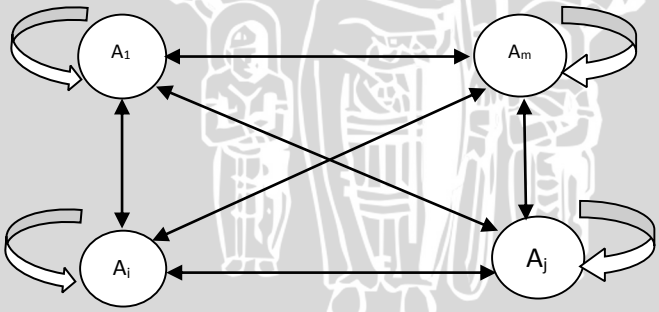


Gambar 2.5 Arsitektur Jaringan Syaraf Tiruan Layer Tunggal

Sumber : (Matondang, 2013)



Gambar 2.6 Arsitektur Jaringan Syaraf Tiruan Multi Layer



Gambar 2.7 Arsitektur Jaringan Syaraf Tiruan Lapisan Kompetitif

2.6.3 *Feed Forward Neural Networks*

Bentuk arsitektur *neural networks* (NN) yang secara umum paling banyak digunakan dalam aplikasi di bidang teknik atau rekayasa adalah *Multi Layer Perceptrons* (MLP) yang juga dikenal dengan *Feed Forward Neural Networks* (FFNN).Fungsi aktivasi yang digunakan pada jaringan *backpropagation* terdiri dari 2 jenis fungsi aktivasi, yaitu fungsi sigmoid biner yang nilainya berkisar antara 0 dan 1 dan fungsi sigmoid bipolar yang nilainya berkisar antara -1 dan 1 (Fernanda & Otok, 2012). Rumus untuk fungsi sigmoid biner diperlihatkan pada persamaan

2.3, sedangkan rumus untuk fungsi sigmoid bipolar diperlihatkan pada persamaan 2.4 berikut ini.

$$y = f(x) = \frac{1}{1+e^{-\alpha x}} \quad \textbf{2.3}$$

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad \textbf{2.4}$$

Langkah-langkah pada *feed forward neural network* adalah sebagai berikut (Fernanda, 2012) :

1. Setiap unit *input* $(x_i, i = 1,\ldots,n)$ akan menerima sinyal *input* x_i dan menyebarkan sinyal yang telah diterima ke semua unit pada *hidden layer*.
2. Setiap unit *hidden* $(z_j, j = 1,\ldots,p)$ harus menjumlahkan bobot dari sinyal *input* dengan menggunakan persamaan 2.5 berikut.

$$z_{inj} = v_{0j} + \sum_{i}^{n} x_i v_{ij} \quad \textbf{2. 5}$$

3. Tahap selanjutnya adalah menerapkan fungsi aktivasi untuk menghitung sinyal *output*, kemudian mengirimkan sinyal tersebut ke semua unit pada *output layer* dengan menggunakan persamaan 2.6.

$$z_j = f(z_{in_j}) \quad \textbf{2.6}$$

4. Setiap unit *output* $(Y_k, k = 1,\ldots,m)$ harus menjumlahkan bobot dari sinyal *input* dengan menggunakan persamaan 2.7 berikut.

$$y_{ink} = w_{0k} + \sum_{j}^{p} z_j w_{jk} \quad \textbf{2. 7}$$

5. tahap selanjutnya adalah menerapkan fungsi aktivasi untuk menghitung sinyal *output* dengan persamaan 2.8

$$y_k = f(y_{in_k}) \quad \textbf{2.8}$$

2.6.4 *Backpropagation*

Model ANN umumnya menggunakan Algoritma *Backpropagation* (BP) yang merupakan pengembangan dari algoritma *least mean square* untuk melatih jaringan dengan beberapa layer. Langkah-langkah metode *backpropagation* adalah sebagai berikut:

1) Setiap unit *output* (y_k, k=1,2,3,…,m) menerima pola target yang sesuai dengan

input pelatihan, kemudian hitung *error* dengan menggunakan persamaan 2.9 berikut.

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad \textbf{2.9}$$

Kemudian hitung suku perubahan bobot menggunakan *learning rate* seperti pada persamaan 2.10.

$$\Delta w_{jk} = (\propto \delta_k z_j) \quad \textbf{2.10}$$

Keterangan:

t_k = target *output*

δ_k = *error* unit *output* y_k ke unit *hidden*

Δw_{jk} = perubahan bobot dari *hidden layer* menuju *output layer*

α = *learning rate*

2) Setiap unit tersembunyi (Z_j, j=1,2,3,…,n) menjumlahkan delta *input*-nya seperti pada persamaan 2.11.

$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk} \quad \textbf{2.11}$$

Mengalikan hasil yang diperoleh dari persamaan 2.11 dengan fungsi aktivasi untuk menghitung informasi *error* menggunakan persamaan 2.12**.**

$$\delta_j = \delta_in_j f(z_in_j) \quad \textbf{2.12}$$

Hitung perubahan bobot menggunakan persamaan 2.13.

$$\Delta v_{ij} = (\propto \delta_j x_i) \quad \textbf{2.13}$$

Keterangan:

δ_j = *error* dari *output layer* ke *hidden layer* z_j

δ_in_j = *error* pada *hidden layer*

Δv_{ij} = perubahan bobot dari *input layer* ke *hidden layer*

Tahapan Perubahan Bobot

1. Setiap unit *output* (y_k, k=1,2,3,…,m) dilakukan perubahan bobot dengan persamaan 2.14.

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad \textbf{2.14}$$

Setiap unit tersembunyi (z_j, j=1,2,3,…,n) dilakukan perubahan bobot dengan persamaan 2.15.

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \quad \textbf{2.15}$$

2.6.5 Proses Normalisasi

Metode normalisasi data yang digunakan dalam penelitian ini adalah

metode *Min-Max*. Normalisasi *Min-Max* merupakan metode yang me-*rescale* data dari suatu *range* ke *range* yang baru. Persamaan matematika dari metode ini adalah sebagai berikut (Chamidah, Wiharto, & Salamah, 2012). Proses normalisasi dilakukan dengan mencari nilai minimal dan maksimal seperti pada persamaan 2.16

$$x_i' = 0.8 * \left(\frac{x_i - \min(x)}{\max(x) - \min(x)}\right) + 0.1 \qquad \textbf{2.16}$$

Keterangan:

x_i' = Nilai normalisasi data ke-*i*

x_i = nilai data ke-*i*

$\min x$ = nilai minimal data

$\max x$ = nilai maksimal data

2.6.6 Proses Denormalisasi

Proses denormalisasi dilakukan untuk mengubah nilai normalisasi ke dalam luaran nilai data asli. Persamaan matematika dari denormalisasi ditunjukkan pada persamaan 2.17 (Gema, 2014) :

$$y_i'' = \frac{((\max(x) - \min(x) * (x_i'' - 0.1))}{0.8} + \min(x) \qquad \textbf{2. 17}$$

Keterangan:

y_i'' = nilai denormalisasi

2.6.7 *Mean Square Error* (MSE)

Mean Square Error (MSE) merupakan salah satu metode yang dapat digunakan untuk melakukan evaluasi kesalahan. Pada ANN, MSE dapat digunakan untuk meningkatkan optimasi dari proses yang dilakukan. Persamaan matematika dari MSE ditunjukkan pada persamaan 2.18 (Edo, 2014) :

$$MSE = \frac{1}{k}\sum_{k=1}^{n}(y_k - t_k)^2 \qquad \textbf{2. 18}$$

Keterangan :

n : jumlah data

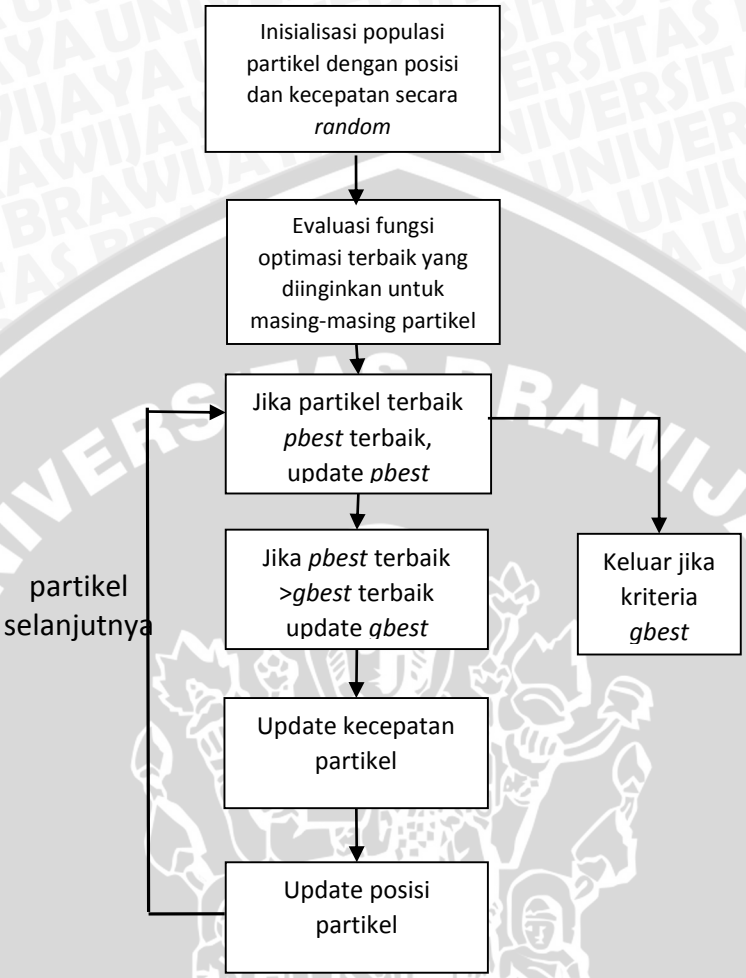
y_k : nilai *output* data

t_k : nilai target data

2.7 Algoritma *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995 sebagai salah satu metode optimasi. *Particle Swarm Optimization* (PSO) merupakan sebuah algoritma pencarian berbasis populasi berdasarkan simulasi perilaku sosial burung dalam sebuah *swarm*. Perilaku sosial

ini terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Dasar prosedur PSO digambarkan pada Gambar 2.8.



Gambar 2.8 Dasar prosedur PSO

Setiap individu berperilaku secara terdistribusi dengan menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi kelompok kolektifnya. Misalnya, jika seekor burung menemukan jalan yang tepat (*optimal*) menuju sumber makanan, maka sisa anggota kelompok yang lainnya juga akan mengikuti jalan tersebut meskipun lokasi mereka didalam kelompok tersebut tidak saling berdekatan. PSO dikembangkan berdasarkan pada model tersebut, dimana ketika partikel mendekati target minimum atau maksimum dari fungsi tujuan, partikel tersebut akan secara cepat mengirim informasi kepada partikel yang lainnya dalam kawanan tertentu sehingga partikel yang lain akan mengikuti arah menuju ke target nilai tersebut tetapi tidak secara langsung. Ada satu komponen yang bergantung pada memori dari setiap partikel yaitu memori tentang apa yang sudah dilalui sebelumnya. Jika jalur tersebut pernah dilalui dan ternyata tidak menghasilkan nilai sesuai fungsi tujuan maka jalur tersebut tidak akan diikuti.

2.7.1 Parameter PSO

Parameter yang dibutuhkan pada algoritma *Particle Swarm Optimization* antara lain (Zerda, 2009) :

1. ***Swarm size***

   Swarm size atau jumlah partikel. Sebenarnya dalam sebagian besar masalah 5 - 10 partikel cukup besar untuk mendapatkan hasil yang bagus. Untuk masalah yang sangat sulit atau khusus, bagus untuk mencoba 100 atau 200 particle.

2. ***Problem dimension***

   Problem dimension atau dimensi solusi dari partikel Ini ditentukan dari masalah yang akan dioptimasi.

3. ***Range***

   Batas dari partikel Ini juga ditentukan dari masalah yang akan dioptimasi. Dapat menspesifikasikan range yang berbeda untuk dimensi yang berbeda dari partikel.

4. ***Inertia weight* atau bobot inersia (w)**

   Bobot inersia dikenalkan oleh Shi dan Eberhart (1998) yang dalam algoritma *Particle Swarm Optimization* digunakan sebagai keseimbangan antara kemampuan eksplorasi global dan lokal secara utama dan merupakan parameter penurunan kecepatan untuk menghindari stagnasi partikel di lokal optimum. Bobot inersia (w) pada dasarnya akan mengontrol seberapa besar pengetahuan dari arah terbang sebelumnya akan mempengaruhi kecepatan baru.

   Nilai w sangat penting untuk memastikan perilaku konvergen dan untuk menukar eksplorasi dan eksploitasi secara optimal. Untuk $w \geq 1$, kecepatan bertambah melampaui waktu dan menuju kecepatan maksimum (diasumsikan dengan menggunakan pembatasan kecepatan) dan *swarm* berubah. Partikel gagal mengubah arah untuk kembali ke posisi yang kemungkinan merupakan solusi. Untuk $w < 1$, kecepatan berkurang hingga mendekati nol. Nilai w yang terlalu kecil akan menghilangkan kemampuan eksplorasi *swarm.* Momentum kecil dari langkah sebelumnya yang menyebabkan perubahan cepat terhadap arah. Nilai w yang lebih kecil akan lebih baik dalam mengontrol dalam pembaruan posisi terbaik lokal ataupun global.

   Inersia akan mengecil sesuai dengan bertambahnya iterasi sehingga kemampuan PSO dalam pencarian lokal akan lebih efisien dan konvergensi ke solusi optimum global, dimana nilai minimum dan maksimum w yang digunakan berada dalam range 0 sampai 1 (Chen, 2011).

5. **Koefisien Kognitif dan Sosial Partikel**

Dalam PSO, pencarian terhadap solusi optimal dipandu oleh dua percepatan komponen yaitu kognitif dan komponen sosial (Engelbrecht A. P., 2007). Oleh karena itu, kontrol yang tepat dari dua komponen ini sangat penting untuk menemukan solusi optimal secara akurat dan efisien. Komponen kognitif (c_1) dan komponen sosial (c_2) disebut juga sebagai koefisien akselerasi yang mempengaruhi jarak maksimum yang dapat diambil oleh sebuah partikel dalam sebuah iterasi. C_1 mengatur jarak yang dipengaruhi oleh memori posisi personal best dari partikel tersebut. Sedangkan c_2 mengatur jarak maksimum yang dipengaruhi oleh partikel global best.

Komponen kognitif yang relatif tinggi dibandingkan dengan komponen sosial akan menghasilkan individu mengembara berlebihan melalui ruang pencarian. Sebaliknya, nilai komponen sosial yang relatif tinggi dapat menyebabkan partikel terburu-buru sebelum waktunya menuju optimum lokal (konvergensi dini). Pertimbangan utama modifikasi ini adalah untuk menghindari konvergensi prematur pada tahap awal dari pencarian dan untuk meningkatkan konvergensi untuk solusi optimum global selama tahap terakhir dari pencarian (A. Ratnaweera, 2004).

6. Kondisi berhenti

Merupakan kriteria yang digunakan untuk mengakhiri iterasi atau perulangan dan mendapatkan solusi optimum. Ada 2 aspek yang dipertimbangkan untuk menentukan kondisi berhenti:

1. Kondisi berhenti tidak boleh menyebabkan PSO konvergen dini karena solusi sub-optimal akan diperoleh.
2. Kondisi berhenti harus menjaga *over sampling* fungsi objektif. Jika kondisi berhenti membutuhkan penghitungan berkala fungsi objektif, kompleksitas komputasi proses pencarian akan meningkat secara signifikan.

   Pilihan atau ketentuan untuk kondisi berhenti yang dapat digunakan untuk mencapai nilai iterasi maksimum, perulangan telah mencapai nilai optimum atau minimum error (*threshold*) yang diinginkan.

2.7.2 Penerapan PSO

PSO memiliki tiga komponen utama yaitu partikel, komponen sosial dan kognitif dari partikel, dan kecepatan partikel. Dalam ruang masalah di mana mungkin ada lebih dari satu solusi sehingga solusi optimal dari masalah diperlukan, partikel merupakan solusi individu untuk masalah. Pembelajaran partikel berasal dari dua sumber, satu dari pengalaman individu partikel yang disebut pembelajaran kognitif dan pembelajaran lainnya adalah pembelajaran gabungan dari seluruh kawanan disebut pembelajaran sosial. Pembelajaran kognitif diwakili oleh individu terbaik (pBest) dan pembelajaran sosial diwakili oleh nilai terbaik secara global (gBest).

Sehingga pada dasarnya algoritma PSO hanya terdiri dari tiga langkah, yang kemudian mengalami perulangan atau iterasi sampai kondisi berhenti yang diharapkan tercapai. Langkah pertama adalah evaluasi nilai fitnes untuk setiap partikel, kemudian memperbarui nilai terbaik individu dan global, dan ketiga adalah memperbarui kecepatan dan posisi setiap partikel.

Berikut merupakan penerapan PSO dalam kasus pemodelan komposisi pakan kambing PE untuk optimasi kandungan gizi susu (Zerda, 2009):

1. Inisialisasi Partikel

Diasumsikan bahwa sebuah partikel untuk setiap dimensinya harus berada dalam domain yang didefinisikan oleh dua vektor, x_{min} dan x_{max} yang mewakili batas bawah dan batas atas setiap dimensi (Engelbrecht, 2007). Ditunjukkan pada persamaan 2.19.

$$x_i(0) = x_{min,d} + r_d\left(x_{max,d} - x_{min,d}\right), \forall d = 1, \dots, n_x, \forall i = 1, \dots, n_s \quad \textbf{2.19}$$

x_{id} adalah posisi partikel ke-i pada dimensi ke-d, dimana $r_j \sim U(0,1)$ merupakan nilai acak (*random*) dengan *range* [0,1]. Inisialisasi posisi dari setiap partikel (X_{id}) dan kecepatan dari setiap partikel (V_{id}).

Kecepatan awal dapat diinisialisasi menjadi nol. Ditunjukkan pada persamaan 2.20.

$$V_{id}(0) = 0 \quad \textbf{2.20}$$

V_{id} kecepatan partikel ke-i pada dimensi ke-d, posisi terbaik individu ke-i $(pBest_i)$ awal untuk setiap partikel diinisialisasikan dengan posisi partikel pada $t = 0$. Ditunjukkan pada persamaan 2.21.

$$pBest_i(0) = x_i(0) \quad \textbf{2.21}$$

2. Fungsi Objektif PSO

Algoritma genetika dapat menyelesaikan masalah kompleks, yaitu permasalahan optimasi. Untuk mendapatkan nilai *fitness*, suatu individu harus dievaluasi berdasarkan suatu nilai fungsi objektif tertentu. Jika permasalahannya adalah optimasi untuk memaksimalkan fungsi h, maka digunakan fungsi objektif $f = h$ untuk mencari nilai *fitness*nya. Akan tetapi, jika permasalahannya adalah optimasi untuk meminimalkan fungsi h, maka digunakan fungsi objektif $f = 1/h$ untuk mencari nilai *fitness*nya. Rumus *fitness* diperlihatkan pada persamaan 2.22 dan 2.23 berikut (Pramesti, Mahmudy, & Indriati, 2015)

$$f = h \quad \textbf{2.22}$$

$$f = \frac{1}{(h+a)} \quad \textbf{2.23}$$

Keterangan:

a = bilangan kecil yang sesuai dengan permasalahan yang dihadapi.

h = fungsi yang akan dimaksimalkan atau diminimalkan.
f = fungsi fitness.

3. Local Best PSO

Local Best PSO dalam kasus minimal optimasi didapatkan dengan perbandingan nilai fitness saat ini dengan nilai pBest sebelumnya, jika fitness saat ini < dari pbest, maka pBest = fitness saat ini (Engelbrecht A. P., 2007). Ditunjukkan pada persamaan 2.25.

$$pBest_i(t+1) = \begin{cases} pBest_i(t) \;\; jika\, f(x_{id}(t+1)) \geq f(pBest_i(t)) \\ x_{id}(t+1) \;\; jika\, f\big(x_{id}(t+1)\big) < f(pBest_i(t)) \end{cases} \quad \textbf{2.24}$$

Keterangan :

$pBest_i(t+1)$ = pBest iterasi sekarang

$pBest_i(t)$ = pBest iterasi sebelumnya

$f(x_{id}(t+1)$= nilai fitness iterasi sekarang

4. Global Best PSO

Pencarian nilai gBest pada kasus minimal optimasi dengan cara mencari nilai minimal semua pBest dalam iterasi *t* (Engelbrecht A. P., 2007). Ditunjukkan pada persamaan 2.26.

$$f(gbest_i(t)) = \; min\{f(x_{id}(t)),\ldots,f(x_{nd}(t+m))\} \quad \textbf{2.25}$$

Keterangan :

$gBest(t)$ = gBest iterasi (t)

$f(x_{id}(t)$= nilai fitness iterasi (t)

5. Update Kecepatan dan Posisi

Kecepatan (V_i) dan posisi dari partikel (X_i) diubah menggunakan persamaan 2.27.

$$V_{id}^{t+1} = w * v_{id}^{t} + c_1 * Rand() * (p_{id} - x_{id}^{t} + c_2 * Rand() * (p_{gd} - x_{id}^{t}) \quad \textbf{2.26}$$

Untuk mengontrol eksplorasi global partikel, perlu adanya pembatasan kecepatan maksimum. Teknik ini disebut velocity clamping untuk mencegah partikel bergerak terlalu jauh melampaui ruang pencariannya. Dalam penelitian ini, batasan kecepatan atau *threshold* yang digunakan ditunjukkan pada persamaan 2.28 (Marini, 2015):

$$jika\;\; v_{id}^{t+1} > v_d^{max} \quad\; maka\, v_{id}^{t+1} = \; v_d^{max} \quad \textbf{2.27}$$
$$jika\;\; v_{id}^{t+1} < -v_d^{max} \quad maka\, v_{id}^{t+1} = -v_d^{max}$$

Strategi velocity clamping telah terbukti lebih efektif dalam meredam osilasi dengan menyediakan keseimbangan yang baik antara eksplorasi global dan eksploitasi local (Fadilah, 2015). Namun dalam menentukan nilai v_d^{max} bergantung pada bentuk permasalahan yang diselesaikan. Pada penelitian ini rumus Vmax ditunjukkan pada persamaan 2.29 (Engelbrecht A. P., 2007).

$$v_{max} = k * (x_{max} - x_{min}) \quad \textbf{2.28}$$

Rumus perubahan posisi (Xi) ditunjukkan pada persamaan 2.30 (Marini, 2015) :

$$x_{id}^{t+1} = x_{id}^{t} + V_{id}^{t+1} \quad \textbf{2.29}$$

Rumus W inersia (w) ditunjukkan pada persamaan 2.31 (Chen, 2011) :

$$w = (w_{max} - w_{min}) * \frac{t_{max} - t}{t_{max}} + w_{min} \quad \textbf{2.30}$$

Dimana $w_{\max}$ dan $w_{\min}$ masing-masing adalah nilai awal dan nilai akhir bobot inersia yang nilainya secara berurutan adalah 0,9 dan 0,4. Sedangkan $i_{\max}$ adalah jumlah iterasi maksimum yang digunakan.

Keterangan :

v_{id}^{t} = kecepatan partikel ke-i pada dimensi ke-d pada iterasi ke-t

x_{id}^{t} = posisi partikel ke-i pada dimensi ke-d pada iterasi ke-t

w = bobot inersia

c_1, c_2 = konstanta akselerasi

p_{id} = P best atau posisi terbaik partikel pada x_{id}^{t}

p_{gd} = P global best atau posisi terbaik partikel secara keseluruhan

6. Random Injection

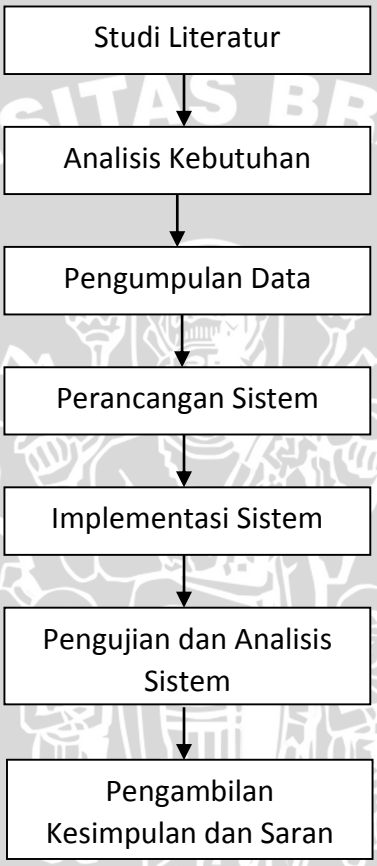
Pada ruang pencarian yang tidak terlalu besar, sering dijumpai pencapaian konvergensi dini. Hal ini disebabkan karena partikel lebih cepat menemukan posisi terbaik global dalam ruang pencarian yang kecil dan disebabkan oleh kurangnya diversitas populasi setelah melewati sekian generasi (Mahmudy W. , 2015). Untuk mengatasi hal tersebut dan membuat partikel lebih teliti atau bertahap dalam melakukakn eksploitasi lokal dan eksplorasi global, maka diterapkan sistem *random injection* (Mahmudy W. , 2015). Random injection dilakukan dengan menginisialisasi kembali posisi n partikel setiap g interval iterasi. Penentuan n dan g yang sesuai dilakukan berdasarkan beberapa percobaan sebelumnya pada sistem PSO.

7. Kondisi berhenti

Jika telah mencapai kondisi akhir (mencapai nilai iterasi maksimum atau perulangan telah mencapai nilai optimum) maka perulangan berhenti dan nilai optimumnya didapatkan namun jika belum maka diulangi pada nomor 2.

BAB 3 METODOLOGI PENELITIAN

Bab ini akan membahas metodologi penelitian dalam penelitian "Pemodelan Komposisi Pakan Kambing Peranakan Etawa (PE) Untuk Optimasi Kandungan Gizi Susu Menggunakan *Artificial Neural Network* (ANN) Dan *Particle Swarm Optimization* (PSO)" Studi Kasus UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Metode ini digunakan untuk menggali informasi dan mengkaji dari berbagai literatur yang mendukung penelitian penulis. Literatur berasal dari buku, jurnal, artikel dan dokumentasi. Pokok bahasan literatur meliputi :

- Bahan Pakan Kambing
- Kandungan nutrisi pakan kambing
- Kandungan gizi susu kambing PE
- Karakteristik susu kambing berkualitas
- Pemrograman bahasa Java

- Teknik pengujian Sistem

3.2 Analisa Kebutuhan

3.2.1 Deskripsi Umum Sistem

Sistem yang akan dibangun merupakan sistem pemodelan komposisi pakan untuk mengoptimalkan kandungan gizi susu kambing peranakn etawa (PE) menggunakan *Artificial Neural Network* dan *Particle Swarm Optimization. Input* sistem berupa kandungan gizi yang ingin dioptimalkan, berat badan dan masa laktasi kambing peranakan etawa tersebut sesuai kebutuhan *user*, kemudian algoritma ANN digunakan untuk memodelkan komposisi pakan yang akan diberikan kepada kambing untuk menghasilkan kandungan gizi yang dimasukkan *user*. Sedangkan algoritma PSO digunakan untuk mengoptimasi pilihan komposisi pakan yang ada. *Output* sistem berupa hasil rekomendasi komposisi pakan kambing PE untuk menghasilkan kandungan gizi susu yang sudah dimodelkan menggunakan ANN berdasarkan keinginan *user*.

3.2.2 Spesifikasi Lingkungan Implementasi

3.2.2.1 *Perangkat Keras*

Perangkat keras yang digunakan dalam pembuatan sistem ini adalah sebagai berikut :

1. Prosesor Pentium(R) Dual Core CPU 2.3GHz
2. RAM 4.00 GB
3. *Harddisk* 500 GB
4. Monitor 14”

3.2.2.2 *Perangkat Lunak*

Perangkat lunak yang digunakan dalam pembuatan sistem ini adalah sebagai berikut :

1. Sistem operasi Windows 8.1 64 bit
2. *Microsoft Office* 2013
3. Netbeans IDE 8.0.2
4. Bahasa pemorgraman Java

3.3 Pengumpulan Data

Teknik pengumpulan data dengan melakukan wawancara terhadap pihak UPT, narasumber pak Rokhim, S.Pt bidang pakan ternak, drh. Jayawulandari dan pak Kayat, S.Pt untuk data uji laboratorium kandungan gizi susu. Hasil dari wawancara akan ditunjukkan pada Lampiran 2. Selain itu juga dari jurnal-jurnal yang terkait.

Data yang dibutuhkan dalam penelitian ini adalah:

1. Data komposisi pakan hijauan untuk kambing
2. Data hasil laboratorium untuk kandungan gizi susu

Analisis kebutuhan data beserta sumber, metode pengumpulan data dan kegunaannya dijelaskan pada Tabel 3.1 Analisis Kebutuhan Data berikut:

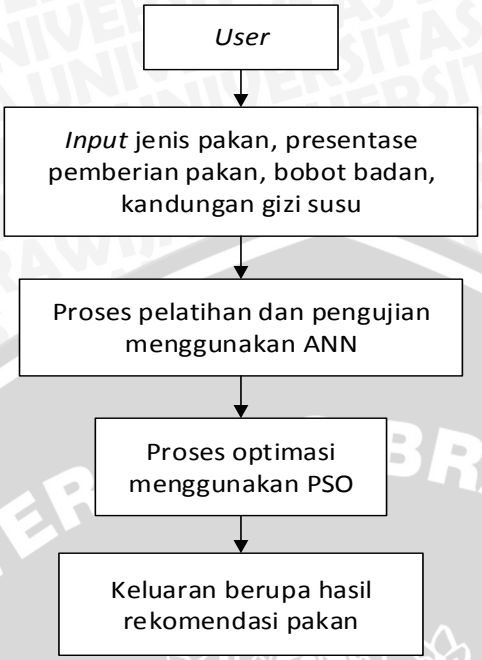
Tabel 3.1 Analisis Kebutuhan Data

No.	Kebutuhan Data	Sumber Data	Metode Pengumpulan Data	Kegunaan Data
1.	Data komposisi pakan hijauan untuk kambing	Jurnal, paper, Pakar pakan Rokhim, S.Pt	Literatur, wawancara	Untuk *input*-an yang digunakan dalam proses *backpropagation*.
2.	Data hasil laboratorium untuk kandungan gizi susu	Pakar drh. Jayawulandari dan Kayat,S.Pt.	wawancara	Digunakan pada proses pelatihan, pengujian *Backpropagaion* sebagai target luaran
3.	Data *range* kandungan gizi susu	Jurnal, paper dan Kayat, S.Pt.	Literatur, wawancara	Digunakan dalam perhitungan PSO untuk optimasi.

3.4 Perancangan Desain Sistem

Berdasarkan analisa kebutuhan data dan implementasi sistem, maka arsitektur perancangan sistem pemodelan optimasi komposisi pakan untuk mengoptimalkan kandungan gizi susu kambing peranakan etawa (PE) dijelaskan dalam Gambar 3.2.

Sistem pemodelan komposisi pakan untuk optimasi kandungan gizi susu kambing ini mengikutsertakan user dalam berjalannya program. User memberikan *input*-an yang berupa jenis rumput dan bobot badan kambing (kg). Jenis rumput ini terdiri dari tiga jenis yaitu, rumput Raja, rumput Odot dan rumput Taiwan. *Input* yang diberikan user akan diproses menggunakan metode ANN dan akan dioptimasi menggunakan metode PSO sehingga didapatkan rekomendasi komposisi pakan agar kandungan gizi susu menjadi optimal.



Gambar 3.2 Perancangan Sistem

3.5 Implementasi Sistem

Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman Java, serta tools pendukung lainnya. *Input* sistem berupa data kebutuhan nutrisi dan pakan kambing, kandungan gizi susu kambing serta parameter algoritma ANN dan PSO. Proses sistem adalah perhitungan komposisi pakan kambing untuk menghasilkan kandungan gizi susu kambing sesuai kebutuhan *user*.

3.6 Pengujian dan Analisis

Tahap pengujian sistem dilakukan untuk menilai keberhasilan dan akurasi sistem yang telah dibuat pada tahap implementasi. Pengujian dilakukan terhadap ukuran populasi yang optimal, banyaknya generasi optimal dan pengujian parameter menggunakan ANN dan PSO berdasarkan dengan skenario pengujian yang telah dibuat.

3.7 Kesimpulan dan Saran

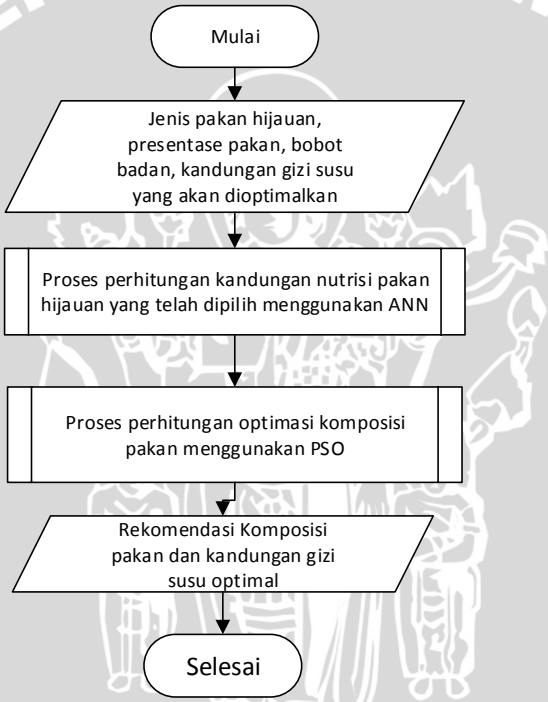
Kesimpulan dilakukan setelah semua tahapan pada metodologi telah selesai dilakukan. Kesimpulan akan mengacu pada hasil pengujian dan analisa sistem untuk menjawab rumusan masalah yang telah ditetapkan. Serta saran untuk memperbaiki kesalahan dan memberikan pertimbangan untuk pengembangan lanjut.

BAB 4 PERANCANGAN

Pada bab ini akan dibahas mengenai diagram alir perancangan system yang akan digunakan, perancangan antarmuka, perancangan pengujian serta manualisasi dari metode yang digunakan.

4.1 Alir Perancangan Sistem

Pada bagian alir perancangan sistem akan dijelaskan rancangan alur kerja sistem dalam bentuk diagram alir untuk menyelesaikan masalah menggunakan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO) secara bertahap. Berikut in merupakan rancangan alir kerja system yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Perancangan Sistem

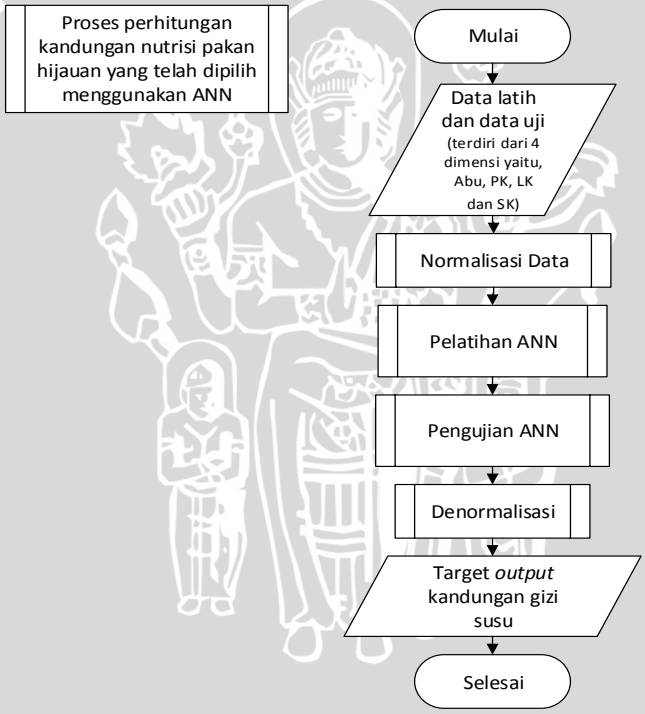
Pada Gambar 4.1 Diagram Alir Perancangan Sistem menggambarkan mengenai proses yang dilakukan sistem dalam pemodelan komposisi pakan kambing PE untuk optimasi kandungan gizi susu menggunakan ANN dan PSO, dijelaskan sebagai berikut :

1. Proses diawali dengan memilih jenis pakan hijauan yang akan dimodelkan dan diberikan pada kambing PE, memasukan bobot badan dan memilih hasil *output* mengenai kandungan gizi susu yang ingin dioptimalkan sesuai dengan keinginan user.

2. Sistem akan melakukan proses penentuan target, bobot dan pengujian dari pelatihan yang akan dilakukan dengan menggunakan ANN sehingga ANN dapat menentukan hasil pemodelan komposisi pakan dengan target kandungan gizi susu yang terbaik berdasarkan jenis pakan hijauan yang dipilih dan bobot badan kambing PE yang dimasukan.
3. Setelah itu, PSO akan mengoptimalkan pengaruh kandungan gizi susu yang dipilih berdasarkan kombinasi kandungan nutrisi bahan pakan hijauan yang telah dipilih sebelumnya.
4. Luaran yang ditampilkan berupa rekomendasi jumlah komposisi pakan dan kandungan gizi yang dihasilkan.

4.1.1 Perhitungan Kandungan Nutrisi Pakan Hijauan Menggunakan ANN

Untuk menghitung kandungan nutrisi pakan hijauan menggunakan ANN, maka pada penelitian ini digunakan *backpropagation neural network* yang ditunjukkan pada Gambar 4.2.



Gambar 4.2 Diagram Alir Proses ANN

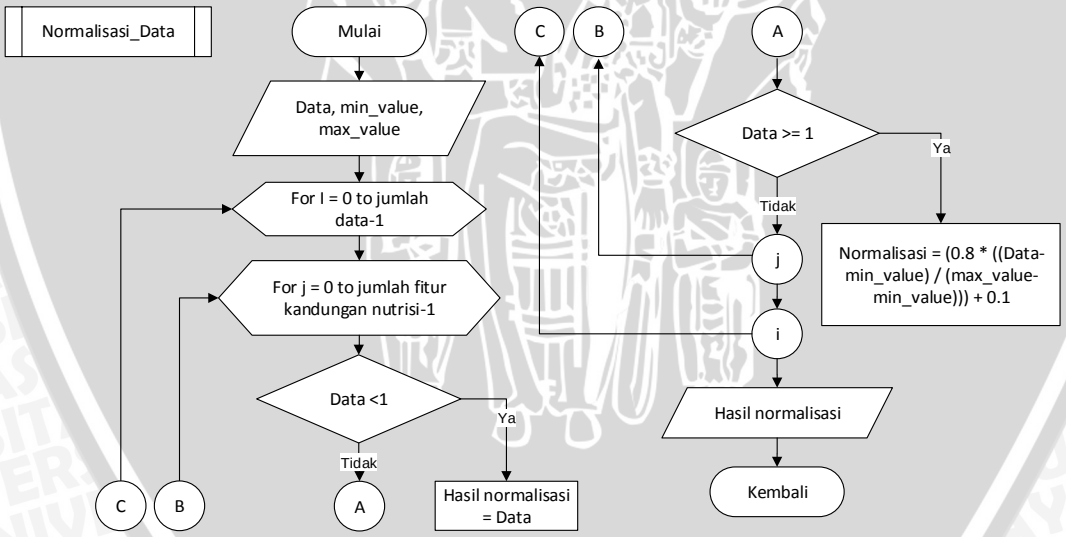
Pada Gambar 4.2 Diagram Alir Proses ANN *Backpropagation* dilakukan dengan tahapan-tahapan sebagai berikut :

1. *Input* data latih dan data uji yang belum dinormalisasi.
2. Proses pertama adalah melakukan normalisasi data. Normalisasi dilakukan untuk data yang nilainya lebih dari 1. Untuk proses normalisasi dihitung dengan Persamaan 2.15.

3. Lakukan pelatihan data latih agar ditemukan bobot terbaik untuk menentukan target atau *output* yang diharapkan dari kandungan gizi susu yang ingin dioptimalkan.
4. Lakukan pengujian pada data uji untuk membuktikan bahwa bobot yang dihasilkan dari pelatihan sudah menghasilkan luaran yang terbaik.
5. Selanjutnya luaran tentang kandungan gizi susu dan data masukkan kandungan nutrisi pakan hijauan akan di denormalisasi untuk mengembalikan nilainya ke rentang semula. Proses denormalisasi dihitung dengan Persamaan 2.16.
6. Diperoleh hasil luaran berupa pemodelan kandungan nutrisi pakan hijauan dengan MSE terkecil.

4.1.1.1 *Proses Normalisasi Data*

Normalisasi data merupakan proses transformasi data dimana nilai atribut data diskalakan menjadi lebih kecil dengan rentang antara 0.1 sampai dengan 0.9. Pada penelitian ini, proses normalisasi data menggunakan metode *Min-Max Normalization*. Normalisasi dilakukan dengan mencari nilai batas maksimal dan minimal dari data yang digunakan. Diagram alir proses normalisasi data ditunjukkan pada Gambar 4.3.



Gambar 4.3 Diagram Alir Proses Normalisasi

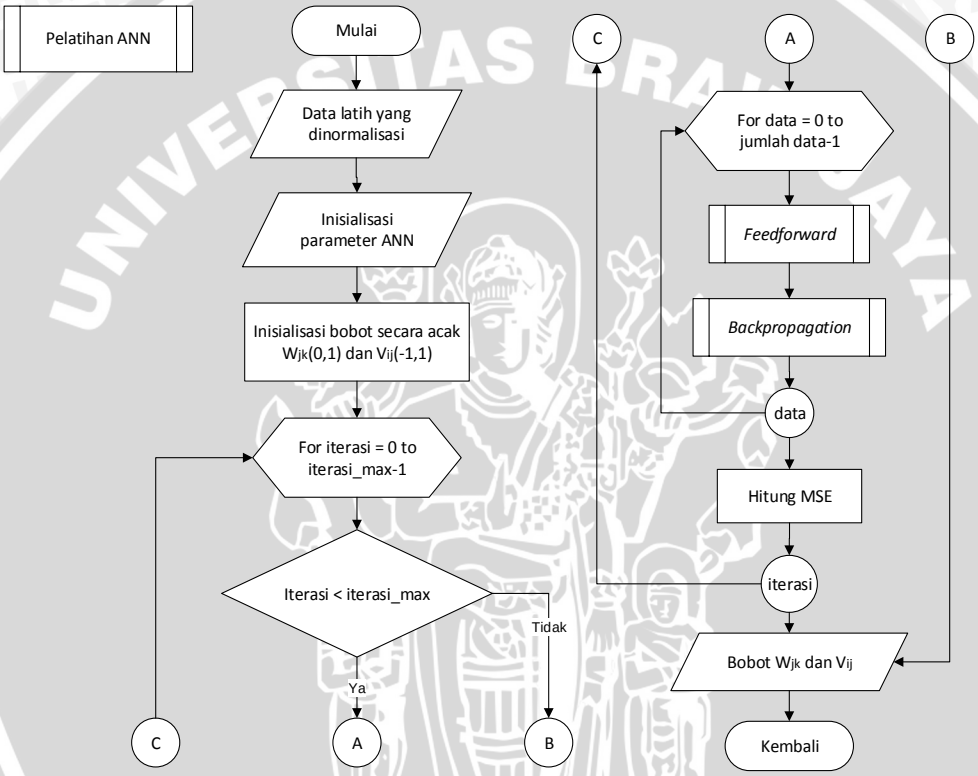
Pada Gambar 4.3 Diagram Alir Proses Normalisasi dilakukan dengan tahapan-tahapan sebagai berikut :

1. Masukan data yang berupa data latih atau data uji yang belum dinormalisasi, *min_value* dan *max_value*.
2. Sistem melakukan pengulangan sebanyak jumlah data, baik data yang berupa data latih ataupun data uji.

3. Sistem melakukan pengulangan sebanyak jumlah fitur kandungan nutrisi bahan pakan seperti abu, PK, LK dan SK.
4. Sistem melakukan percabangan, jika nilai datanya < 1, maka data tersebut tidak akan melalui proses normalisasi data. Sedangkan, jika nilai datanya > =1, maka data tersebut akan dinormalisasikan menggunakan Persamaan 2.15.
5. Sistem akan menghasilkan hasil normalisasi dari data latih ataupun data uji.

4.1.1.2 *Proses Pelatihan ANN*

Berikut adalah diagram alir yang menjelaskan tentang proses pelatihan ANN secara bertahap, ditunjukkan pada Gambar 4.4.



Gambar 4.4 Diagram Alir Proses Pelatihan ANN

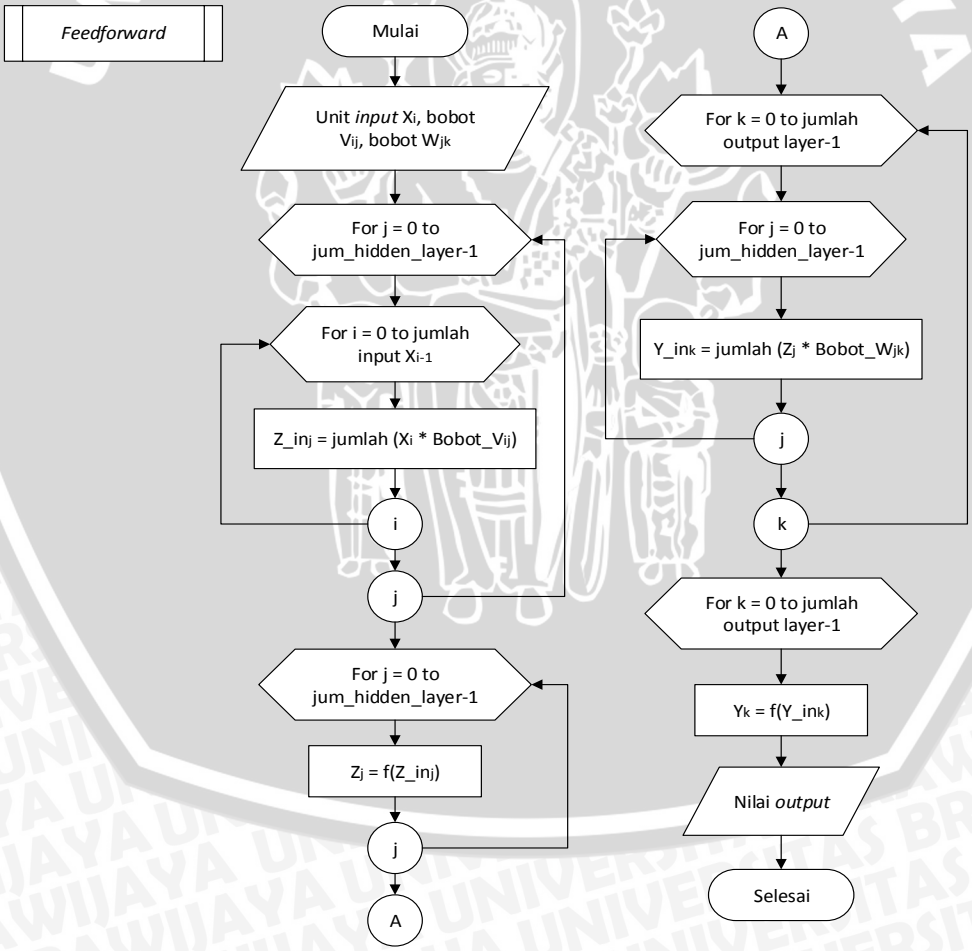
Pada Gambar 4.4 Diagram Alir Proses Pelatihan ANN dilakukan dengan tahapan-tahapan sebagai berikut :

1. Masukan data latih yang telah melewati proses normalisasi data.
2. Masukan parameter inisialisasi parameter yang dimiliki oleh ANN *Backpropagation* yang berupa jumlah iterasi maksimal, jumlah varian *hidden layer*, nilai *learning rate,* jumlah data latih dan berat badan kambing.
3. Sistem membangkitkan bobot *neuron* secara *random*, yaitu bobot *neuron input* dibangkitkan antara *range* -1 sampai 1 dan bobot *neuron output* dibangkitkan antara *range* 0 sampai 1.
4. Sistem melakukan pengulangan sebanyak iterasi_max yang ditetapkan.

5. Sistem melakukan percabangan, jika iterasi masih < iterasi_max atau MSE > galat_max, maka lakukan langkah selanjutnya. Sedangkan, jika iterasi atau MSE sudah tercapai, maka lakukan langkah ke-9.
6. Sistem melakukan pengulangan sebanyak jumlah data yang dimasukan.
7. Sistem melakukan proses *feedforward* dan proses *Backpropagation*.
8. Sistem menghitung nilai MSE dengan menggunakan Persamaan 2.18.
9. Sistem menghasilkan *output* berupa bobot v_{ij} dengan Persamaan 2.14 dan dan w_{jk} yang terakhir dengan Persamaan 2.13.

4.1.1.3 *Proses Feedforward pada ANN*

Pada proses *feedforward*, setiap unit *input* akan menerima sebuah masukan sinyal dan meneruskan ke setiap *hidden layer*. Setiap *hidden layer* menghitung nilai aktivasi dan mengirimkan sinyal ke setiap unit *output*. Setiap unit *output* kemudian menghitung aktivasinya untuk memberikan pola masukan. Berikut adalah diagram alir dari proses *feedforward* pada ANN yang ditunjukkan pada Gambar 4.5.



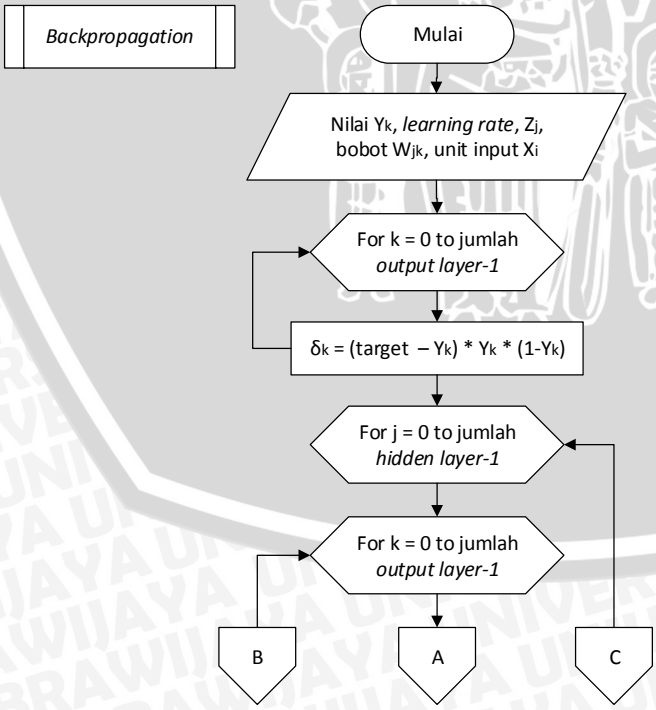
Gambar 4.5 Diagram Alir Proses *Feedforward* pada ANN

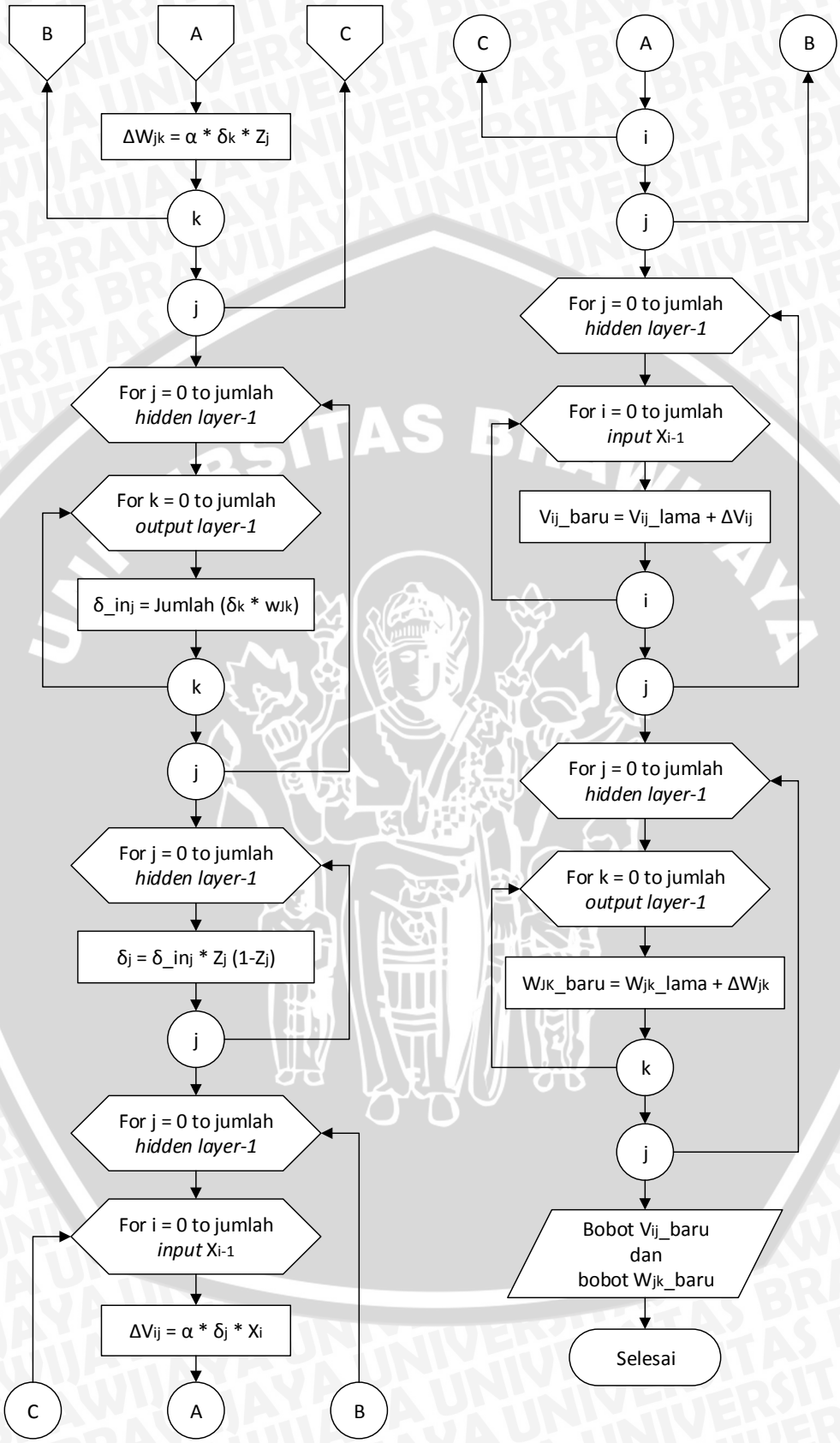
Pada Gambar 4.5 Diagram Alir Proses *Feedforward* pada ANN dilakukan dengan tahapan-tahapan sebagai berikut :

1. Masukan unit *input* X_i sebagai data *input*, bobot V_{ij} dan bobot W_{jk}.
2. Sistem melakukan pengulangan sebanyak jumlah *hidden layer* dan pengulangan sebanyak jumlah *input* X_i yang telah dimasukan sebelumnya.
3. Sistem menjumlahkan bobot sinyal *input* dengan melakukan proses perhitungan terhadap nilai Z_in_j dengan menggunakan Persamaan 2.4.
4. Sistem melakukan pengulangan sebanyak jumlah *hidden layer*.
5. Sistem menghitung fungsi aktivasi sigmoid biner pada *hidden layer* (Z_j) dengan menggunakan Persamaan 2.5.
6. Sistem melakukan pengulangan sebanyak jumlah *output layer* yang telah ditetapkan sebelumnya dan pengulangan sebanyak jumlah *hidden layer*.
7. Sistem menjumlahkan bobot sinyal *output* dengan melakukan proses perhitungan terhadap nilai Y_in_k dengan menggunakan Persamaan 2.6.
8. Sistem melakukan pengulangan sebanyak jumlah *output layer*.
9. Sistem menghitung fungsi aktivasi sigmoid biner pada *output layer* (Y_k) dengan menggunakan Persamaan 2.7.
10. Sistem menghasilkan nilai *output* dari proses *feedforward* seperti nilai Z_in_j, Z_j, Y_in_k dan Y_k.

4.1.1.4 *Proses Backpropagation pada ANN*

Berikut adalah diagram alir yang menjelaskan tentang proses *Backpropagation* ANN secara bertahap yang ditunjukkan pada Gambar 4.6.





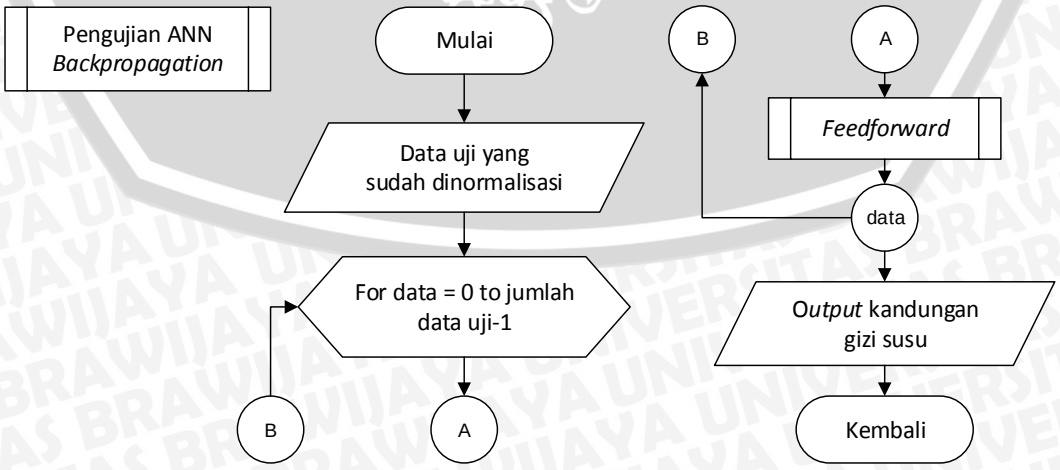
Gambar 4.6 Diagram Alir Proses *Backpropagation* pada ANN

Pada Gambar 4.6 Diagram Alir Proses *Backpropagation* pada ANN dilakukan dengan tahapan-tahapan sebagai berikut :

1. Masukan nilai Y_k, *learning rate*, Z_j, bobot W_{jk} dan *input* X_i sebagai data *input*.
2. Sistem melakukan pengulangan sebanyak jumlah *output layer* kemudian menghitung *error output* dengan menggunakan Persamaan 2.8.
3. Sistem melakukan pengulangan sebanyak jumlah *hidden layer* dan pengulangan sebanyak jumlah *output layer*.
4. Sistem akan melakukan proses perhitungan perubahan bobot pada *output layer* dengan menggunakan Persamaan 2.9.
5. Lakukan langkah ke-3, kemudian menjumlahkan setiap *error* pada *output layer* terhadap *hidden layer* dengan menggunakan Persamaan 2.10.
6. Sistem melakukan pengulangan sebanyak jumlah *hidden layer* dan menghitung informasi *error* dengan menggunakan Persamaan 2.11.
7. Sistem melakukan pengulangan sebanyak jumlah *hidden layer* dan pengulangan sebanyak jumlah data *input* (*input* X_i).
8. Sistem menghitung perubahan bobot pada *hidden layer* dengan menggunakan Persamaan 2.12.
9. Lakukan langkah ke-7, kemudian menghitung bobot V_{ij} yang baru dengan menggunakan Persamaan 2.13.
10. Lakukan langkah ke-3, kemudian menghitung bobot W_{jk} yang baru dengan menggunakan Persamaan 2.14.
11. Sistem menghasilkan *output* berupa bobot V_{ij}_baru dan W_{jk}_baru.

4.1.1.5 *Proses Pengujian ANN Backpropagation*

Setelah proses pelatihan ANN *Backpropagation* selesai dilakukan, langkah selanjutnya adalah melakukan proses pengujian ANN *Backpropagation*. Berikut adalah diagram alir yang menjelaskan tentang proses pengujian ANN *Backpropagation* yang ditunjukkan pada Gambar 4.7.



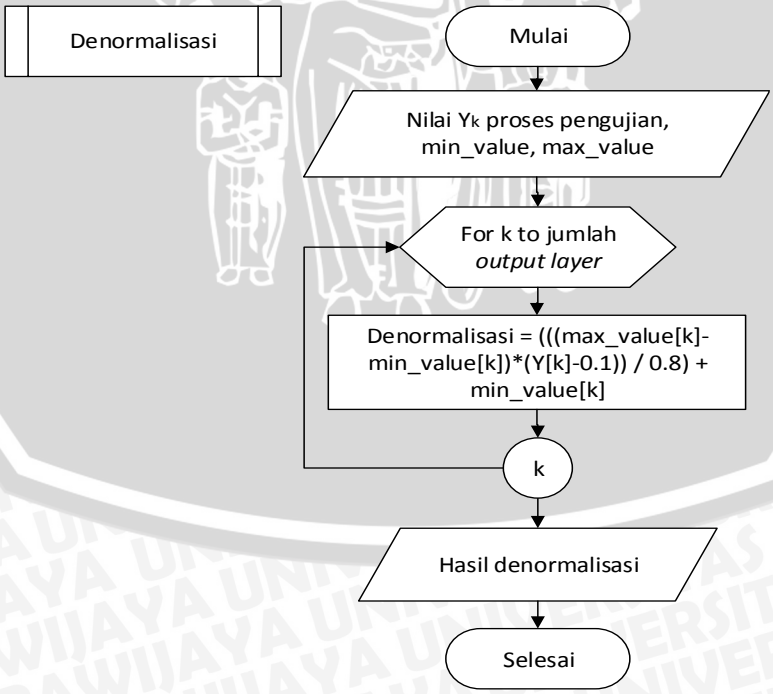
Gambar 4.7 Diagram Alir Proses Pengujian ANN *Backpropagation*

Proses pengujian ANN *Backpropagation* hanya melakukan tahap *feedforward* saja. Seluruh bobot *input* yang digunakan pada proses pengujian diambil dari nilai bobot terakhir yang diperoleh dari proses pelatihan ANN *Backpropagation*. Pada tahap pengujian ini, jaringan diharapkan mampu mengenali pola berdasarkan data baru yang diberikan. Pada Gambar 4.7 Diagram Alir Proses Pengujian ANN *Backpropagation* dilakukan dengan tahapan-tahapan sebagai berikut :

1. *Input* data uji yang sudah dinormalisasi.
2. Lakukan pengulangan sebanyak jumlah data uji yang dimasukan.
3. Setiap data uji akan melakukan proses *feedforward* yang telah dijelaskan pada sub bab 4.2.1.3.
4. Setelah pengulangan selesai, langkah selanjutnya adalah sistem menampilkan hasil luaran kandungan gizi susu pada setiap data uji berdasarkan kandungan nutrisi yang dimiliki oleh salah satu jenis pakan hijauan yang telah dipilih.

4.1.1.6 *Proses Denormalisasi*

Proses denormalisasi dilakukan setelah proses pengujian ANN *Backpropagation* telah selesai dilakukan. Denormalisasi dilakukan untuk mengembalikan nilai kandungan gizi susu kambing PE ke dalam nilai asli. Perubahan tersebut dilakukan guna mempermudah dalam pemberian takaran kandungan nutrisi setiap komposisi pakan yang telah dipilih sebelumnya. Berikut adalah diagram alir yang menjelaskan tentang proses denormalisasi yang ditunjukkan pada Gambar 4.8.



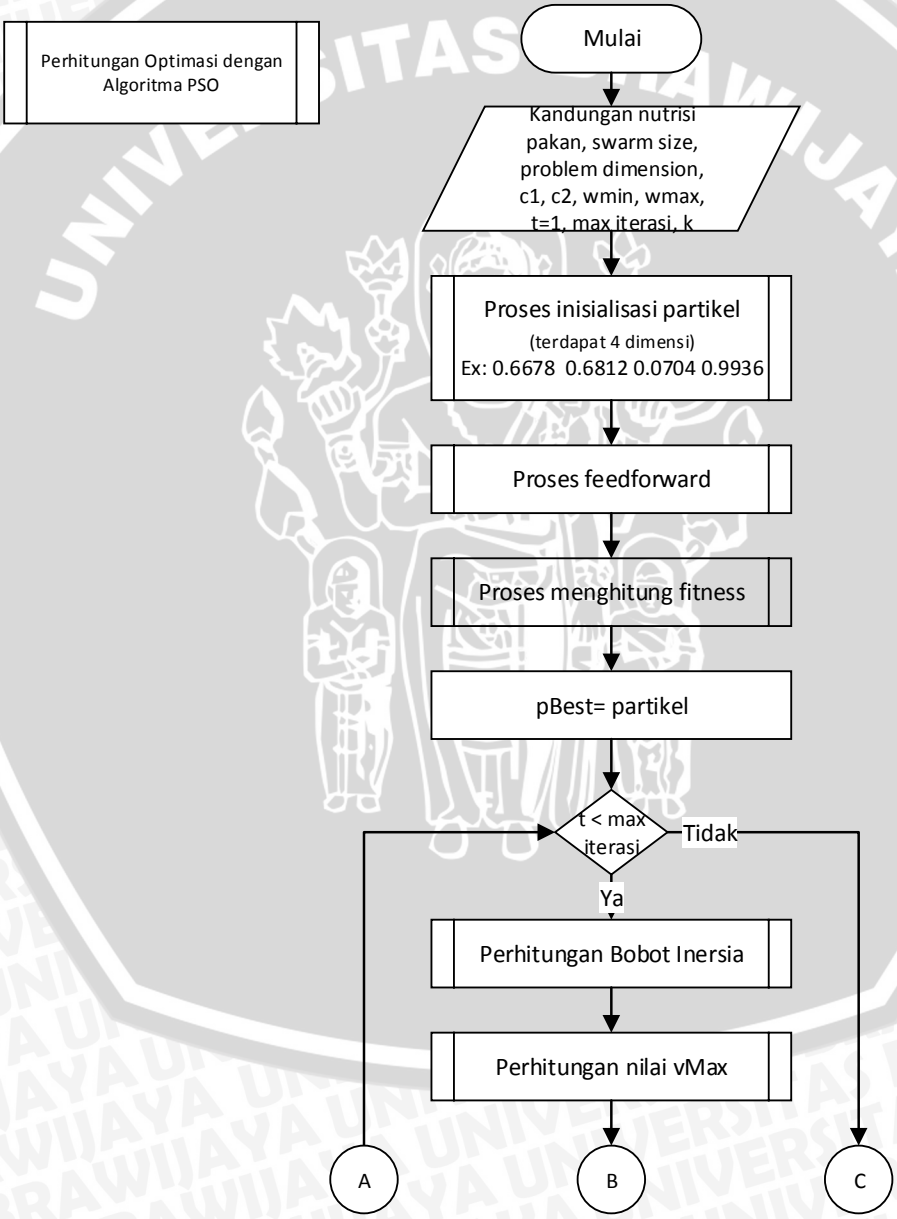
Gambar 4.8 Diagram Alir Proses Denormalisasi

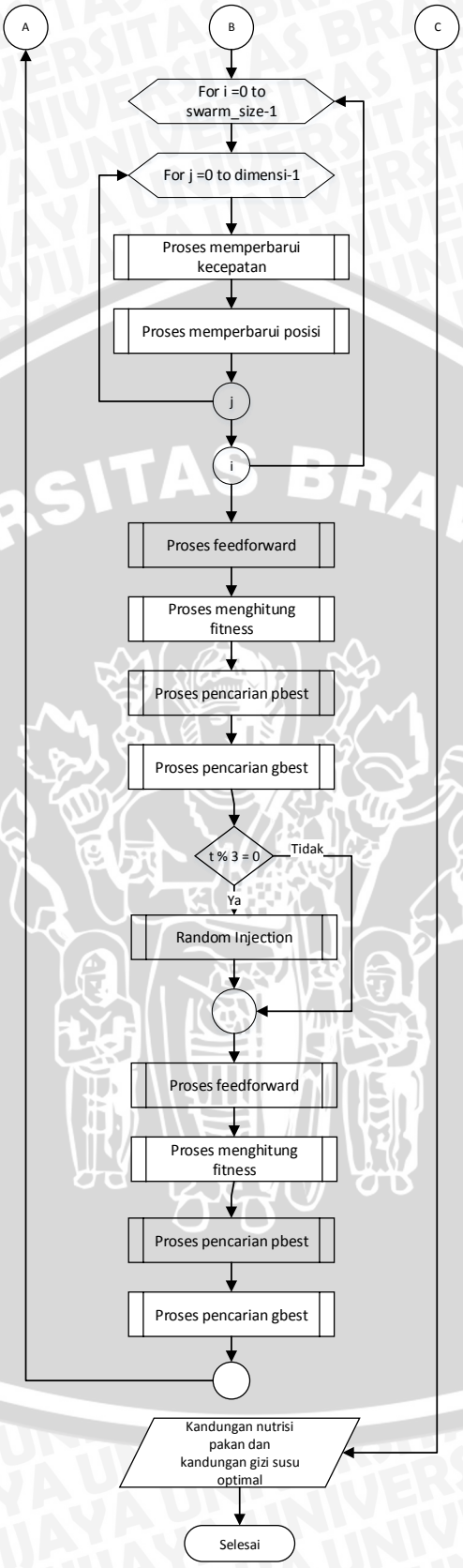
Pada Gambar 4.8 Diagram Alir Proses Denormalisasi dilakukan dengan tahapan-tahapan sebagai berikut :

1. Masukan nilai Y_k dari hasil proses penujian ANN *Backpropagation*, *min_value* dan *max_value*.
2. Sistem melakukan pengulangan sebanyak jumlah *output layer* dan menghitung denormalisasi data dengan menggunakan Persamaan 2.16.
3. Sistem menghasilkan *output* hasil perhitungan denormalisasi data.

4.1.2 Perhitungan Optimasi Komposisi Pakan Menggunakan PSO

Untuk mengahsilkan output sistem yang diinginkan, dilakukan proses penghitungan menggunakan algoritma PSO dengan diagram alir seperti ditunjukkan pada Gambar 4.9 berikut ini.





Gambar 4.9 Diagram alir proses algoritma PSO

Adapun proses algoritma PSO yang ditunjukkan pada Gambar 4.9 dapat dijabarkan sebagai berikut:

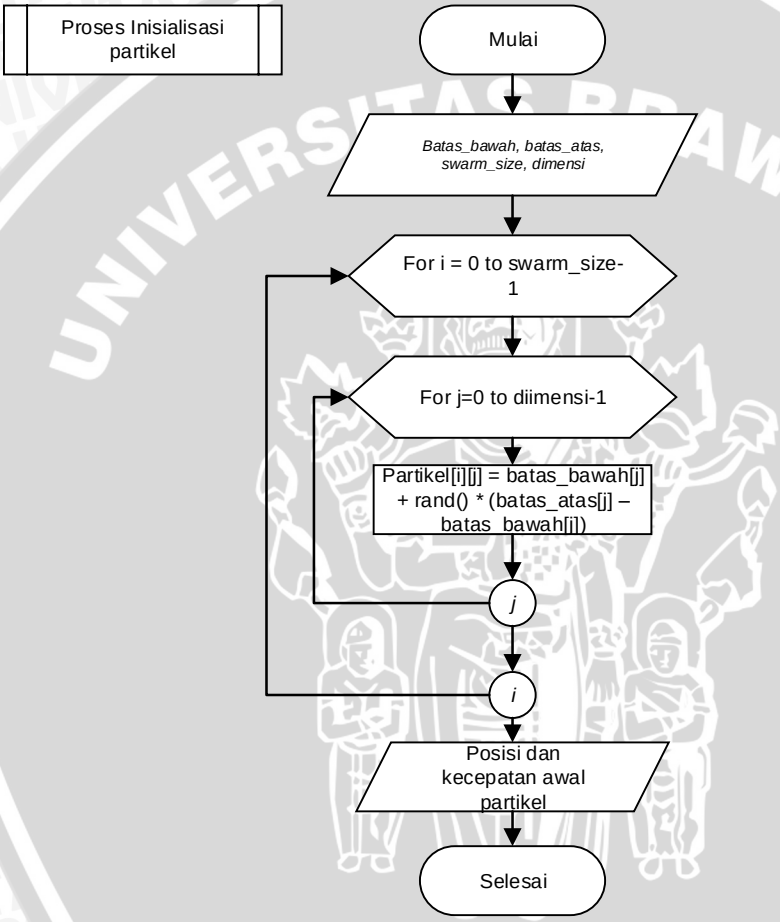
1. Memasukkan parameter yang dibutuhkan dalam algoritma PSO
2. Proses inisialisasi partikel yaitu membangkitkan posisi partikel.
3. Proses menghitung pemodelan pakan menggunakan *feedforward.*
4. Proses menghitung fungsi obyektif PSO (*fitness*).
5. Untuk partikel awal atau iterasi(t) = 0 , maka *pBest* = partikel seperti pada Persamaan 2.15, belum terjadi proses berbandingan.
6. Membangkitkan perulangan *for* dengan kondisi $t < max\ iterasi$, jika kondisi bernilai *true* akan berlanjut ke langkah memperbarui *pBest*, jika kondisi bernilai *false* maka akan langsung keluar dari perulangan menuju hasil.
7. Menghitung nilai bobot inersia (w) dengan menggunakan Persamaan 2.20.
8. Proses untuk memperbarui kecepatan partikel dengan menggunakan Persamaan 2.18 dengan acuan v_{max} dengan Persamaan 2.19.
9. Proses untuk memperbarui posisi partikel dengan menggunakan Persamaan 2.20.
10. Dari hasil posisi baru partikel maka dilakukan pemodelan dari partikel menggunakan *feedforward* kemudian dilakukan penghitungan nilai fitness. Dari hasil nilai fitness inilah yang digunakan untuk memperbarui *pBest*.
11. Proses memperbarui *pBest* dengan menggunakan Persamaan 2.16.
12. Proses mencari *gBest* dengan menggunakan Persamaan 2.17.
13. Jika iterasi merupakan kelipatan 3 makan dilakukan proses *random injection*.
14. Dari hasil posisi baru partikel maka dilakukan pemodelan dari partikel menggunakan *feedforward* kemudian dilakukan penghitungan nilai fitness. Dari hasil nilai fitness inilah yang digunakan untuk memperbarui *pBest*.
15. Setelah keluar dari perulangan *for*, maka akan keluar hasil perhitungan algoritma PSO.

4.1.3 Proses inisialisasi partikel

Proses pertama dalam algoritma PSO adalah inisialisasi partikel. Diasumsikan bahwa sebuah partikel untuk setiap dimensinya harus berada dalam domain yang didefinisikan oleh dua vektor, x_{min} dan x_{max} yang mewakili batas bawah dan batas atas setiap dimensi. Secara garis besar proses dapat dilihat pada Gambar 3.5 berikut.

Berdasarkan diagram allir pada Gambar 4.10, proses inisialisasi partikel dapat dijelaskan sebagai berikut:

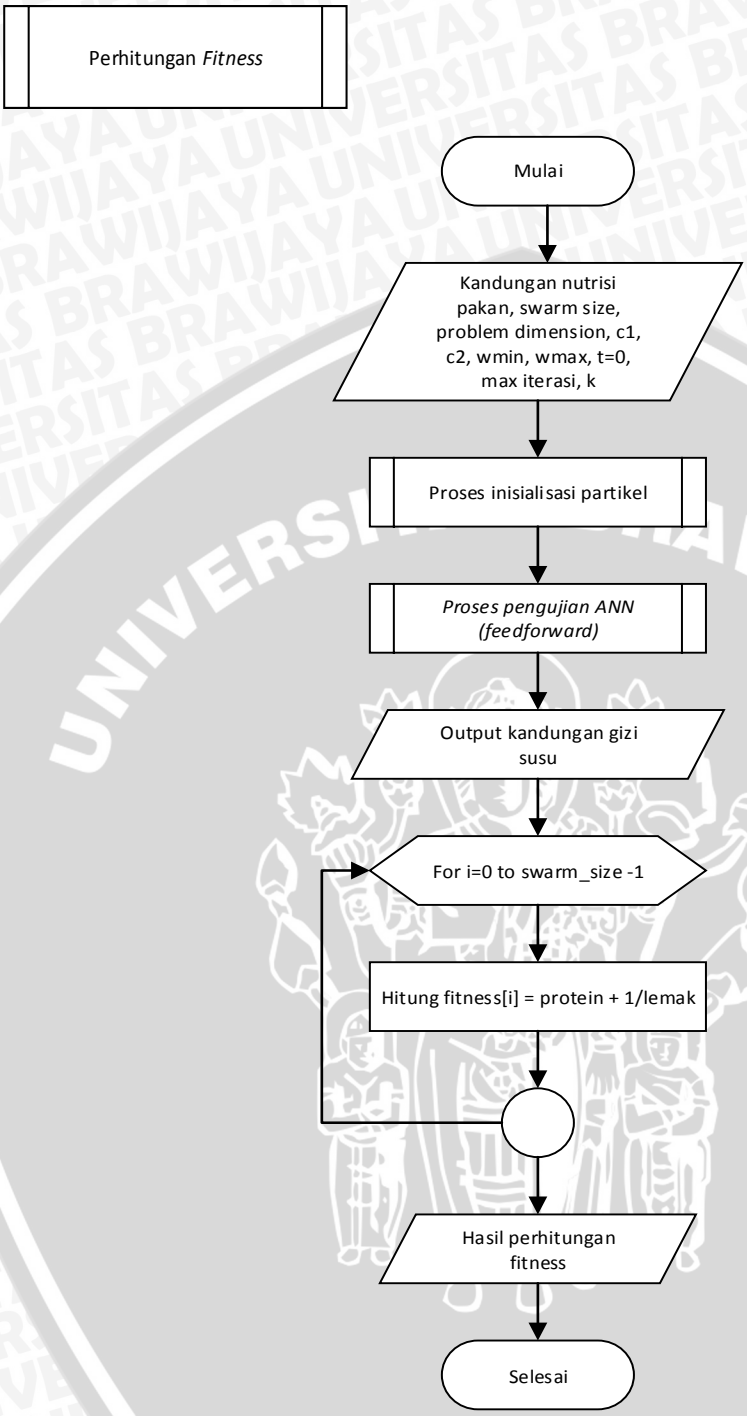
1. Membangkitkan perulangan dengan memberi nilai $i = 0$ sampai pada *swarmsize.*
2. Membangkitkan perulangan dengan memberi nilai $j = 0$ sampai pada *problem dimension.*
3. Kecepatan awal dapat diinisialisasi menjadi 0 sesuai persamaan 2.16.
4. Menentukan posisi partikel dengan Persamaan 2.17.
5. Setelah keluar dari perulangan akan terbentuk partikel yang memiliki posisi dan kecepatan.



Gambar 4.10 Diagram alir proses inisialisasi partikel

4.1.4 Proses menghitung fungsi obyektif PSO

Dalam penelitian ini algoritma PSO digunakan untuk medapatkan perhitungan nilai *fitness* yang paling optimal , yaitu nilai yang dapat melebihi target dari pengujian ANN, semakin besar nilai fitness, semakin bagus hasil kandungan gizi yang dihasilkan. Perhitungan *fitness* menggunakan Persamaan 2.25, namun lebih jelasnya digambarkan melalui diagram alir pada Gambar 4.11.



Gambar 4.11 Diagram alir proses menghitung fungsi obyektif

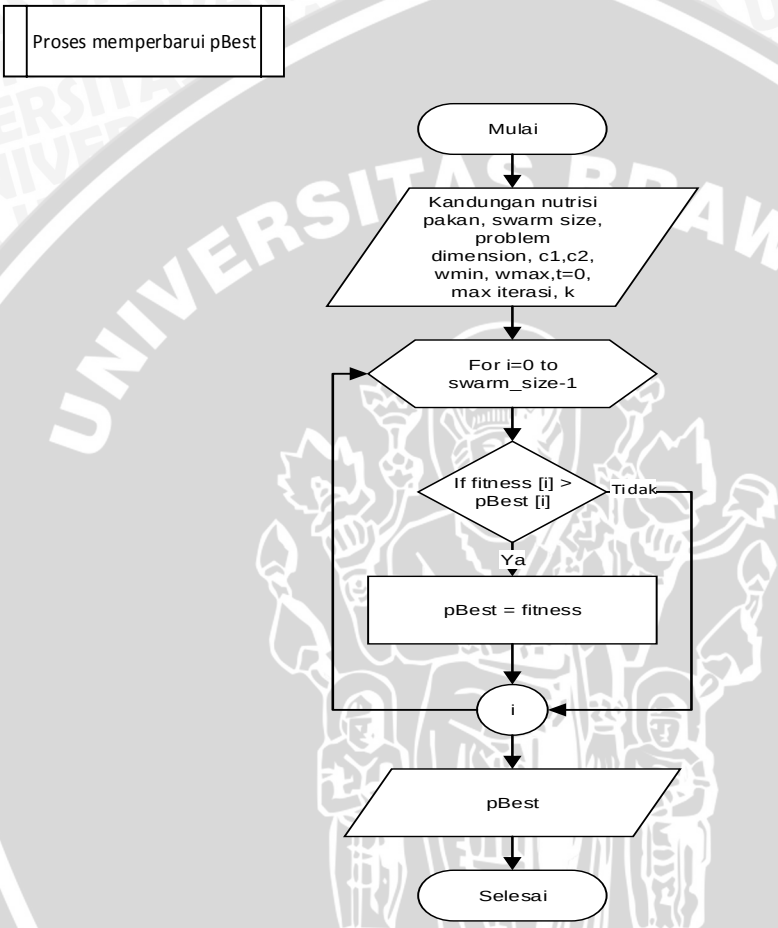
Keterangan dari Gambar 4.11 adalah sebagai berikut:

1. Memasukkan parameter yang dibutuhkan dalam algoritma PSO
2. Proses inisialisasi partikel yaitu membangkitkan posisi partikel.
3. Proses menghitung pemodelan pakan menggunakan *feedforward* yang akan menghasilkan *output* dari hasil denormalisasi.

4. Membangkitkan perulangan dengan memberi nilai $i = 0$ sampai pada *swarmsize.*
5. Proses menghitung fungsi obyektif PSO (*fitness*).

4.1.5 Proses memperbarui *pBest*

Ketentuan dalam Proses memperbarui *pBest* sesuai dengan Persamaan 2.19 ,untuk lebih jelasnya proses tersebut dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram alir proses memperbarui pBest

Proses memperbarui *pBest* adalah proses untuk mencari posisi terbaik setiap individu atau partikel. Proses ini dimulai dengan cara membandingkan nilai dari hasil perhitungan fungsi obyektif kandungan susu optimal terbaik $(f(pBest))$ dan nilai *kandungan susu optimal* partikel iterasi saat ini $(f(x+1))$.

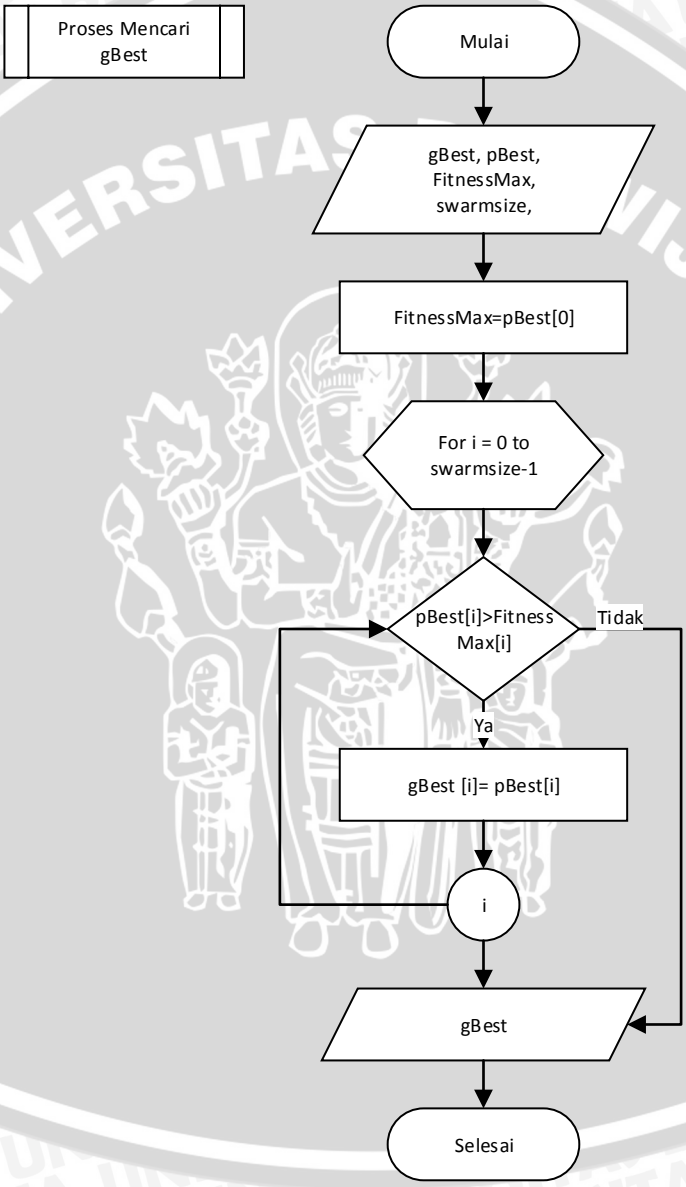
Keterangan Gambar 4.12 adalah sebagai berikut:

1. Memasukkan parameter yang dibutuhkan pada PSO.
2. Membangkitkan perulangan i=0 sampai pada *swarmsize.*
3. Jika *fitness*[i] lebih dari *pBest*[i] maka pBest adalah *fitness*. Jika tidak *pBest* adalah *pBest* itu sendiri.

4. Memperbarui posisi *pBest.*

4.1.6 Proses mencari *gBest*

Sedangkan nilai terbaik global ($gBest$) berasal dari hasil *social learning* dari *swarm* yang ditemukan sebagai posisi individu terbaik dari keseluruhan partikel yang pernah ditemukan. Mencari $gBest$ untuk kasus minimalisasi dihitung sesuai dengan Persamaan 2.20. Untuk lebih jelasnya, proses akan dijelaskan pada Gambar 4.13 berikut ini.



Gambar 4.13 Diagram alir proses mencari gBest

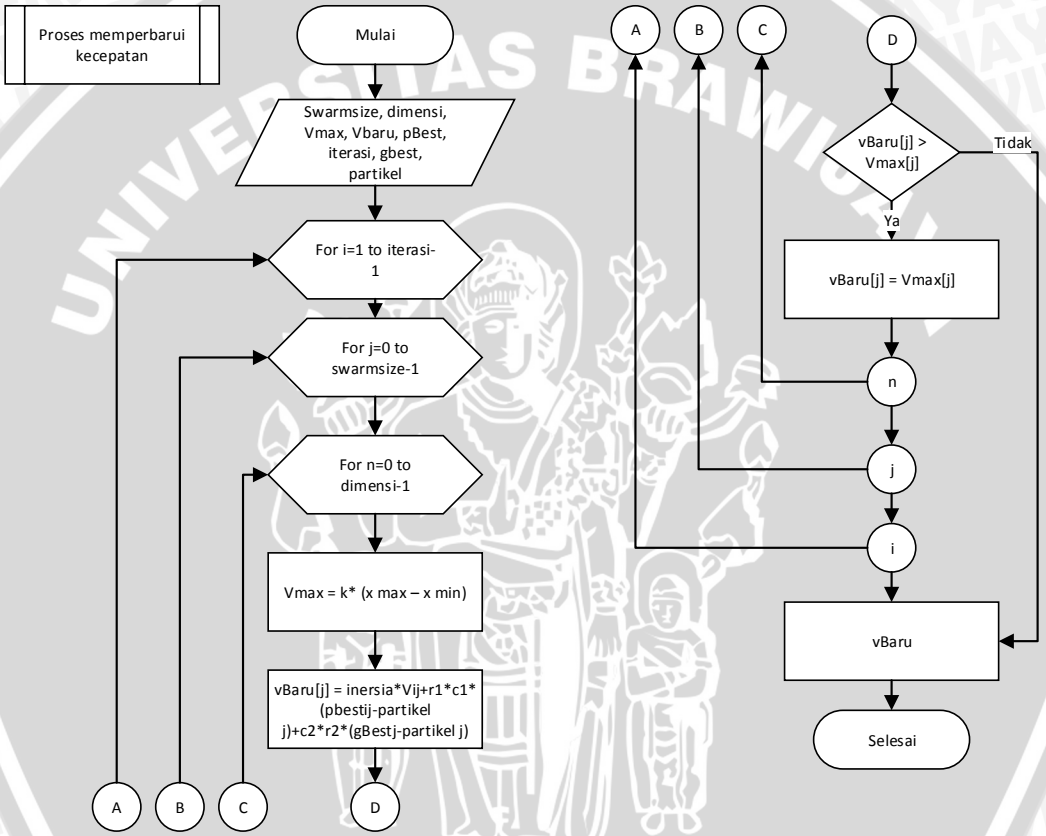
Keterangan dari Gambar 4.13 adalah sebagai berikut:

1. Data yang dibutuhkan dalam proses *gBest* adalah *pBest*, *gBest* dan *FitnessMax.*
2. Membangkitkan perulangan i=0 sampai pada *swarmsize.*

3. Jika *pBest*[i] lebih dari *FitnessMax*[i], maka gBest adalah pBest. Jika tidak, maka *FitnessMax* adalah *gBest.*

4.1.7 Proses memperbarui kecepatan partikel

Berdasarkan $pBest$ dan $gBest$ yang diperoleh, maka kecepatan (v_{id}) dan posisi dari particle (x_{id}) diubah. Proses untuk memperbarui kecepatan partikel dengan menggunakan Persamaan 2.21 dengan acuan v_{max} dengan Persamaan 2.22. Jika hasil perhitungan kecepatan lebih dari v_{max} , maka kecepatan baru sama dengan v_{max} tersebut. Proses untuk memperbarui kecepatan partikel dijelaskan dalam diagram alir pada Gambar 4.14 berikut ini.



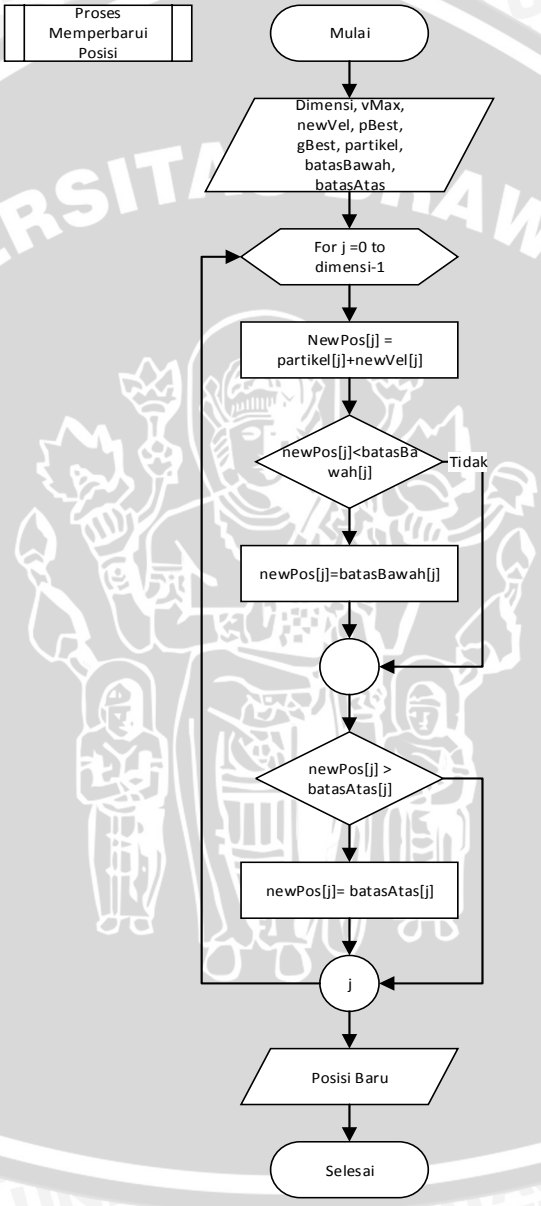
Gambar 4.14 Diagram alir proses memperbarui kecepatan partikel

Keterangan Gambar 4.14 adalah sebagai berikut:

1. Data yang diperlukan dalam update kecepatan.
2. Membangkitkan perulangan *i*=1 sampai pada iterasi, perulangan *j*=0 sampai pada swarmsize, perulangan *n*=0 sampai pada dimensi.
3. Menghitung nilai *Vmax*
4. Menghitung kecepatan baru
5. Jika vBaru lebih dari *vMax* maka vBaru adalah *vMax*. Jika tidak maka cetak vBaru.

4.1.8 Proses memperbarui posisi partikel

Hasil dari proses memperbarui kecepatan adalah kecepatan baru partikel yang akan ditambahkan pada posisi partikel sekarang untuk mendapatkan posisi baru partikel, dengan kata lain individu bergerak menuju posisi baru sesuai dengan kecepatan. Persamaan untuk perubahan posisi (x_{id}) dijelaskan pada Persamaan 2.23, sedangkan diagram alir proses memperbarui posisi partikel akan dijelaskan dalam Gambar 4.15 berikut ini.



Gambar 4.15 Diagram alir proses memperbarui posisi partikel

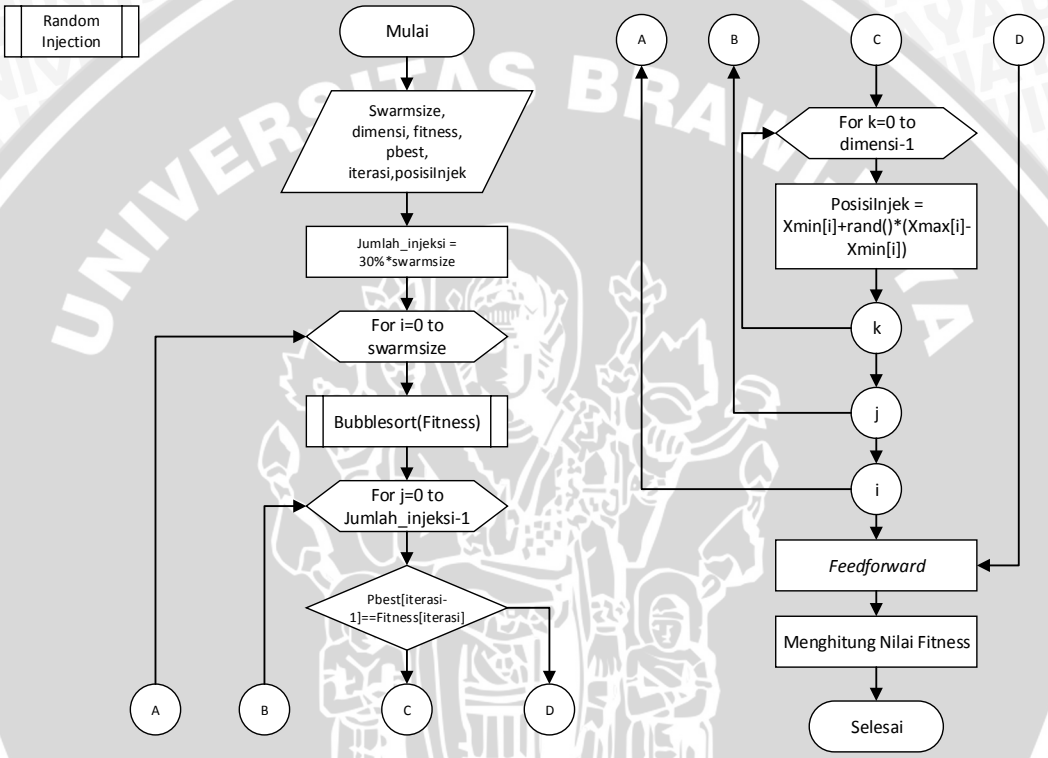
Keterangan Gambar 4.15 adalah sebagai berikut:

1. Data yang dibutuhkan dalam memperbarui posisi
2. Membangkitkan perulangan *j*=0 sampai pada dimensi.

3. Perhitungan *NewPos*[*j*].
4. Jika *NewPos* kurang dari *batasBawah[j]*, maka *newPos* adalah *batasBawah[j]*.
5. Jika *newPos* lebih dari *batasAtas[j]* maka *newPos* adalah *batasAtas[j].*

4.1.9 Proses *Random Injection*

Proses *random injection* dilakukan jika iterasi adalah kelipatan 3, jika nilai pbest iterasi *t*-1 sama dengan *fitness* iterasi *t* maka posisi akan diacak, sedangkan diagram alir proses memperbarui posisi partikel akan dijelaskan dalam Gambar 4.16.



Gambar 4.16 Diagram alir proses *random injection*

Berikut merupakan keterangan dari Gambar 4.16:

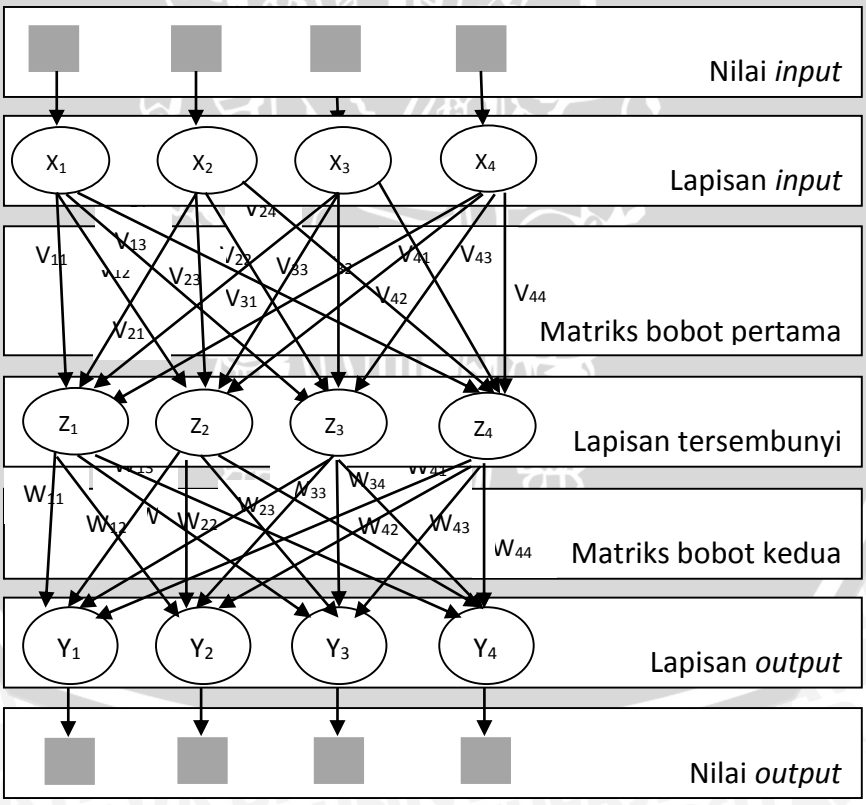
1. Data yang dibutuhkan pada *random injection*.
2. Membangkitkan perulangan *i*=0 sampai pada *swarmsize*.
3. Proses *sorting* nilai *fitness*.
4. Membangkitkan perulangan sebanyak jumlah_injeksi.
5. Jika *pBest[iterasi-t]* memiliki nilai yang sama dengan *Fitness*[*iterasi*] maka dilakukan perulangan sampai pada dimensi.
6. Perhitungan posisi yang diinjeksi.
7. Jika tidak, maka dilakukan proses *feedforward*.
8. Menghitung nilai *fitness*.

4.2 Perhitungan Manualisasi ANN *Backpropagation*

4.2.1 Analisis Permasalahan

Dalam ANN *Backpropagation* menyelesaikan kasus dengan menentukan nilai bobot terbaik sehingga mendapatkan target output terbaik atau mendekati kandungan gizi susu dengan nilai error yang kecil. Setiap jenis rumput memiliki kandungan nutrisi yang berbeda-beda, dan kandungan nutrisi tersebut yang dijadikan sebagai data input dalam perhitungan manual. Ada empat kandungan gizi susu yang diamati, yaitu protein, lemak, laktosa dan berat jenis. Terdapat arsitektur jaringan *Multilayer Net* yang menggambarkan layer yang digunakan dalam kasus optimasi kandungan gizi susu pada Gambar 4.17.

Pada proses perhitungan manualisasi untuk proses pelatihan *backpropagation* diberikan jumlah data latih sebanyak sembilan data, kemudian dilakukan perhitungan sampai iterasi kedua, bobot terakhir dari iterasi kedua akan digunakan dalam pengujian data uji dengan banyak data sebanyak enam data yang diambil dari data latih. Hasil bobot terakhir dari pengujian ANN akan digunakan sebagai bobot dalam perhitungan PSO.



Gambar 4.17 Arsitektur Jaringan Kasus Optimasi Kandungan Gizi Susu

Dalam menyelesaikan kasus pemodelan pakan kambing untuk optimasi kandungan gizi susu menggunakan Artificial Neural Network dan Particle Swarm Optimization terdapat empat data input yang merupakan kandungan nutrisi pakan

yaitu Abu, PK, LK dan SK. Data output yang digunakan adalah empat kandungan gizi susu yaitu protein, lemak, laktosa dan *density.* Pada proses manualisasi, berat badan kambing yang digunakan adalah 32 kg dengan dua jenis pakan rumput, yaitu rumput Odot dan rumput Raja yang memiliki prosentase kombinasi 70%:30%.

4.2.2 Proses Normalisasi Data Latih

Pada perhitungan ini akan dilakukan normalisasi untuk data latih dan data uji. Sampel data latih kandungan nutrisi pakan hijauan rumput odot dan rumput raja beserta kandungan gizi susu pada penelitian ini berjumlah 9 data dengan berat badan kambing sebesar 32 kg. Data kandungan nutrisi rumput Odot dan rumput Raja terdapat pada Tabel 4.1.

Tabel 4.1 Kandungan Nutrisi rumput Odot dan rumput Raja

Kandungan Nutrisi	Rumput Odot	Rumput Raja
Abu	0.1697	0.186
PK	0.1703	0.135
LK	0.0176	0.035
SK	0.2484	0.2484

Dalam perhitungan manualisasi, pakan yang digunakan adalah dua jenis pakan yaitu rumput Odot dan rumput Raja, sehingga kandungan nutrisi masing-masing rumput akan dijumlahkan dan total kandungan nutrisi kedua rumput ditunjukkan pada Tabel 4.2.

$$Total\ Abu = Abu_{Odot} + Abu_{Raja}$$

$$Total\ Abu = 0.1697 + 0.186 = 0.3557$$

Tabel 4.2 Total Kandungan Rumput

Kandungan Nutrisi	Odot dan Raja
Abu	0.3557
PK	0.3053
LK	0.0526
SK	0.4968

Data hasil kandungan rumput akan digunakan dalam mendapatkan nilai dari data input pada data latih sebelum normalisasi. Kandungan rumput pada Tabel 4.1 merupakan kandungan rumput yang terdapat pada masing-masing rumput dalam satu kilogram. Sehingga total kandungan rumput pada Tabel 4.2 merupakan kandungan rumput dengan berat dua kilogram yang terdiri dari rumput Odot yaitu satu kilogram dan rumput Raja adalah satu kilogram. Untuk mengetahui jumlah kandungan nutrisi yang terdapat pada pakan kambing, maka jumlah pakan (kg) dengan menggunakan Persamaan 2.1 untuk batas bawah dan Persamaan 2.2 untuk batas atas dikalikan dengan total kandungan rumput yang digunakan. Berikut contoh perhitungan nilai batas bawah dan batas atas kandungan rumput.

$BatasBawah_{Abu} = 10\% * 32 * 0.3557 = 1.1382$

$BatasAtas_{Abu} = 12\% * 32 * 0.3557 = 1.3659$

Tabel batas bawah kandungan dan batas atas kandungan terdapat pada Tabel 4.3.

Tabel 4.3 Batas Bawah dan Batas Atas Kandungan Pakan

Kandungan Nutrisi	Batas Bawah	Batas Atas
Abu	1.1382	1.3659
PK	0.9770	1.1724
LK	0.1683	0.2020
SK	1.5898	1.9077

Dalam memperoleh nilai pada data input dikalikan dengan nilai *random* dalam *range* batas bawah sampai batas atas kandungan sesuai Persamaan 2.19. Berikut perhitungan data input dengan nilai *random*.

$Data\ input_Abu_1 = 1.1382 + \text{rand}() * (1.3659\text{-}1.1382) = 1.2940$

Berikut adalah tabel yang menjelaskan tentang data latih kandungan nutrisi bahan pakan hijauan rumput odot dan rumput raja beserta kandungan gizi susunya sebelum normalisasi pada Tabel 4.4.

Tabel 4.4 Data Latih Sebelum Normalisasi

No	Nama Perlakuan	Kandungan Nutrisi				Output			
		Abu	PK	LK	SK	Protein	Lemak	Laktosa	Density
1	Premium	1.2940	1.0570	0.1810	1.7340	4.11	7.18	3.93	28.49
2	Premium	1.3170	1.0010	0.1920	1.7270	4.08	7.07	3.89	28.95
3	Premium	1.3010	1.0350	0.1980	1.7150	3.83	8.19	3.67	25.33
4	Premium	1.2890	1.1630	0.1800	1.9070	3.97	6.52	3.79	27.8
5	Premium	1.1840	1.1220	0.1800	1.8160	4.13	7.48	3.94	28.36
6	Premium	1.3090	1.1000	0.1850	1.6710	3.89	6.25	3.14	22.3
7	Premium	1.3340	0.9890	0.1920	1.7710	3.84	7.5	3.67	25.93
8	Premium	1.1920	1.0430	0.1860	1.6540	3.81	7.45	3.64	25.71
9	Premium	1.2360	1.0020	0.1710	1.6530	3.83	6.72	3.65	26.45

Adapun sampel data uji kandungan nutrisi bahan pakan hijauan rumput odot dan rumput raja yang digunakan pada penelitian ini berjumlah 6 data. Berikut adalah tabel yang menjelaskan tentang data uji kandungan nutrisi bahan pakan hijauan rumput odot dan rumput raja sebelum normalisasi pada Tabel 4.5.

Tabel 4.5 Data Uji Sebelum Normalisasi

No	Nama Perlakuan	Kandungan Nutrisi			
		Abu	PK	LK	SK
1	Premium	1.2940	1.0570	0.1810	1.7340

2	Premium	1.3170	1.0010	0.1920	1.7270
3	Premium	1.3010	1.0350	0.1980	1.7150
4	Premium	1.2890	1.1630	0.1800	1.9070
5	Premium	1.1840	1.1220	0.1800	1.8160
6	Premium	1.3090	1.1000	0.1850	1.6710

Dari data pada Tabel 4.4 diketahui minimal dan maksimal untuk *input* dan *output* ditunjukkan pada Tabel 4.6.

Tabel 4.6 Tabel Minimal dan maksimal

Diketahui :			
Min(Abu)=	1.184	Min (Protein)=	3.79
Max(Abu)=	1.334	Max (Protein)=	8.7
Min(PK)=	0.989	Min (Lemak)=	6.25
Max(PK)=	1.163	Max (Lemak)=	13.59
Min(LK)=	0.171	Min (Laktosa)=	3.14
Max(LK)=	0.198	Max (Laktosa) =	8.29
Min(SK)=	1.653	Min (BJ) =	22.3
Max(SK)=	1.907	Max (BJ) =	61.45

Proses normalisasi data pada Tabel 4.4 Data Latih Sebelum Normalisasi dan Tabel 4.5 Data Uji Sebelum Normalisasi dilakukan pada abu, PK, LK, SK, protein, lemak, laktosa dan *density*. Berikut adalah hasil normalisasi untuk data latih pada Tabel 4.7 Data Latih Sesudah Normalisasi.

Tabel 4.7 Data Latih Sesudah Normalisasi

No	Nama Perlakuan	Kandungan Nutrisi				Output			
		Abu	PK	LK	SK	Protein	Lemak	Laktosa	Density
1	Premium	0.6867	0.4126	0.3963	0.3551	0.8500	0.4835	0.8900	0.8447
2	Premium	0.8093	0.1552	0.7222	0.3331	0.7750	0.4381	0.8500	0.9000
3	Premium	0.724	0.3115	0.9	0.2953	0.1500	0.9000	0.6300	0.4645
4	Premium	0.66	0.9	0.3667	0.9	0.5000	0.2113	0.7500	0.7617
5	Premium	0.1	0.7115	0.3667	0.6134	0.9	0.6	0.9	0.8
6	Premium	0.7667	0.6103	0.5148	0.1567	0.3000	0.1	0.1	0.1
7	Premium	0.9	0.1	0.7222	0.4717	0.1750	0.6155	0.6300	0.5367
8	Premium	0.1427	0.3483	0.5444	0.1031	0.1	0.5948	0.6000	0.5102
9	Premium	0.3773	0.1598	0.1	0.1	0.1500	0.2938	0.6100	0.5992

Pada Tabel 4.6 diketahui *min value* dan *max value* untuk kandungan nutrisi abu dari sembilan data adalah 1.1840 dan 1.3340, sehingga contoh perhitungan normalisasi untuk kandungan nutrisi abu didapatkan 0.6867 untuk perlakuan pertama yaitu Premium dengan data asli kandungan nutrisi abu adalah 1.2940. Contoh perhitungan normalisasinya sebagai berikut :

Normalisasi kandungan abu = (0.8*(1.2940-1.1840)/(1.3340-1.1840)) + 0.1

= 1.2940

Berikut adalah hasil proses normalisasi data untuk data uji yang diperlihatkan pada Tabel 4.8 Data Uji Sesudah Normalisasi dengan melakukan perhitungan seperti yang dicontohkan sebelumnya.

Tabel 4.8 Data Uji Sesudah Normalisasi

No	Nama Perlakuan	Kandungan Nutrisi			
		Abu	PK	LK	SK
1	Premium	0.6867	0.4126	0.3963	0.3551
2	Premium	0.8093	0.1552	0.7222	0.3331
3	Premium	0.7240	0.3115	0.9000	0.2953
4	Premium	0.6600	0.9000	0.3667	0.9000
5	Premium	0.1000	0.7115	0.3667	0.6134
6	Premium	0.7667	0.6103	0.5148	0.1567

4.2.2.1 *Perhitungan Pelatihan Data Latih ANN*

Proses pelatihan ANN digunakan untuk melatih data latih yang tersedia ke dalam jaringan untuk dapat melakukan perubahan bobot hingga diperoleh nilai MSE yang terendah. Pada kasus permasalahan ini, proses pelatihan ANN dilakukan pada 9 data latih dengan iterasi sebanyak dua kali. Berikut adalah proses manualisasi untuk iterasi pertama terhadap data latih :

1. Iterasi Pertama

Pada iterasi pertama, iterasi dimulai dari t=1. Iterasi pertama pada ANN merupakan langkah awal dari pembangkitan kecepatan (V_{ij}) antara bobot dari unit *input* (i) menuju unit *hidden* (j) secara *random*.

1.1 Perhitungan *Feedforward*

a. Proses *Input Layer* ke *Hidden Layer*

Proses *input layer* ke *hidden layer* (V_{ij}) dibangkitkan secara acak untuk data pertama dari (-1,1). Jumlah *hidden layer* yang digunakan pada manualisasi adalah 4 *hidden layer*. Berikut adalah data ke-1 dari perlakuan Premium sesudah normalisasi pada iterasi pertama yang diperlihatkan pada Tabel 4.9 dan bobot *input layer* ke *hidden layer* diperlihatkan pada Tabel 4.10 sebagai berikut :

Tabel 4.9 Data ke-1 Perlakuan Premium Sesudah Normalisasi

No	Nama Perlakuan	Kandungan Nutrisi			
		Abu	PK	LK	SK
1	Premium	0.6867	0.4126	0.3963	0.3551

Tabel 4.10 Bobot *Input Layer* ke *Hidden Layer*

V_{ij}	Z_1	Z_2	Z_3	Z_4
X_1	0.7105	0.2525	-0.0664	0.4017
X_2	-0.5918	-0.0935	-0.3608	-0.5565
X_3	-0.6764	0.1381	0.7026	-0.0508
X_4	-0.1384	-0.7931	-0.6888	-0.3314

Bobot untuk data ke-2 sampai ke-9 didapatkan pada proses perubahan bobot data yang sebelumnya. Hanya data ke-1 pada iterasi pertama saja yang bobotnya dibangkitkan secara acak.

b. Menjumlahkan Bobot Sinyal *Input*

Langkah selanjutnya adalah menjumlahkan bobot sinyal *input*. Contoh manualisasi perhitungan bobot sinyal *input* dari data ke-1 perlakuan premium sebagai berikut :

$$Z_in_1 = (0.6867 * 0.7105) + (0.4126 * -0.5918) + (0.3963 * -0.6764) + (0.3551 * -0.1384) = -0.0735$$

$$Z_in_2 = (0.6867 * 0.2525) + (0.4126 * -0.0935) + (0.3963 * 0.1381) + (0.3551 * -0.7931) = -0.0921$$

$$Z_in_3 = (0.6867 * -0.0664) + (0.4126 * -0.3608) + (0.3963 * 0.7026) + (0.3551 * -0.6888) = -0.1606$$

$$Z_in_4 = (0.6867 * 0.4017) + (0.4126 * -0.5565) + (0.3963 * -0.0508) + (0.3551 * -0.3314) = -0.0916$$

Berikut adalah hasil perhitungan penjumlahan bobot sinyal *input* yang didapatkan dari 9 data latih yang telah dinormalisasi pada Tabel 4.11 Jumlah Z_in_j Bobot Sinyal *Input* sebagai berikut :

Tabel 4.11 Jumlah Z_in_j Bobot Sinyal *Input*

Z_in_1	Z_in_1	Z_in_3	Z_in_4
-0.0735	-0.0921	-0.1606	-0.0916
-0.0512	0.0287	0.1676	0.0921
-0.3230	0.0472	0.2661	-0.0282
-0.4586	-0.5951	-0.7455	-0.5658
-0.7156	-0.4980	-0.4472	-0.5982
-0.2210	0.0650	-0.0379	-0.1299
-0.0611	-0.1254	0.0222	0.0443
-0.5411	-0.0460	0.1381	-0.2386
0.0494	-0.0197	-0.1112	-0.0074

c. Menerapkan Fungsi Aktivasi Menghitung Sinyal *Input*

Fungsi aktivasi yang digunakan untuk menyelesaikan permasalahan ini adalah fungsi aktivasi sigmoid biner. Berikut adalah contoh manualisasi proses perhitungan sinyal *input* :

e = bilangan *euler*, yaitu 2.71828

$$Z_1 = \frac{1}{(1+2.71828^{-(-0.0735)})} = 0.4816$$

$$Z_2 = \frac{1}{(1+2.71828^{-(-0.0921)})} = 0.4470$$

$$Z_3 = \frac{1}{(1+2.71828^{-(-0.1606)})} = 0.4599$$

$$Z_4 = \frac{1}{(1+2.71828^{-(-0.0916)})} = 0.4771$$

Berikut adalah hasil sinyal *input* dengan menerapkan fungsi aktivasi sigmoid biner yang didapatkan dari 9 data latih yang telah dinormalisasi pada Tabel 4.12 Hasil Perhitungan Sinyal *Input* dengan Fungsi Sigmoid Biner sebagai berikut :

Tabel 4.12 Hasil Perhitungan Sinyal *Input* dengan Fungsi Sigmoid biner

Z_1	Z_2	Z_3	Z_4
0.4816	0.4770	0.4599	0.4771
0.4872	0.5072	0.5418	0.5230
0.4199	0.5118	0.5661	0.4929
0.3873	0.3555	0.3218	0.3622
0.3284	0.3780	0.3900	0.3548
0.4450	0.5162	0.4905	0.4676
0.4847	0.4687	0.5055	0.5111
0.3679	0.4885	0.5345	0.4406
0.5123	0.4951	0.4722	0.4982

d. Bobot dari *Hidden Layer* ke *Output Layer*

Untuk data ke-1, bobot dari *hidden layer* ke *output layer* dibangkitkan secara *random* dengan *range* (0,1). Sedangkan, data ke-2 sampai data ke-9 ditentukan berdasarkan perubahan bobot data yang sebelumnya. Berikut adalah bobot dari *hidden layer* ke *output layer* untuk data ke-1 pada Tabel 4.13 Bobot *Hidden Layer* ke *Output Layer* sebagai berikut :

Tabel 4.13 Bobot *Hidden Layer* ke *Output Layer*

W_{11}	W_{21}	W_{31}	W_{41}	W_{12}	W_{22}	W_{32}	W_{42}
0.5716	0.4962	0.1278	0.1989	0.9383	0.6603	0.7164	0.7722

W_{13}	W_{23}	W_{33}	W_{43}	W_{14}	W_{24}	W_{34}	W_{44}
0.3699	0.7822	0.1866	0.7479	0.5031	0.3135	0.8484	0.3829

e. Menjumlahkan Setiap Bobot Sinyal *Output*

Langkah selanjutnya adalah menjumlahkan setiap bobot sinyal *output*. Contoh manualisasi perhitungan bobot sinyal *output* dari data ke-1 perlakuan Premium sebagai berikut :

$$Y_{in_1} = (0.4816 * 0.5716) + (0.4770 * 0.9383) + (0.4599 * 0.3699) + (0.4771*0.5031) = 1.1330$$

$$Y_{in_2} = (0.4816 * 0.4962) + (0.4770 * 0.6603) + (0.4599 * 0.7822) + (0.4771*0.3135) = 1.0633$$

$$Y_{in_3} = (0.4816 * 0.1278) + (0.4770 * 0.7164) + (0.4599 * 0.1866) + (0.4771*0.8484) = 0.8939$$

$$Y_{in_4} = (0.4816 * 0.1989) + (0.4770 * 0.7722) + (0.4599 * 0.7479) + (0.4771*0.3829) = 0.9908$$

Berikut adalah hasil menjumlahkan bobot sinyal *output* yang didapatkan dari 9 data latih yang telah dinormalisasi ketika iterasi pertama yang diperlihatkan pada Tabel 4.14 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer* sebagai berikut :

Tabel 4.14 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer*

Y_in_1	Y_in_2	Y_in_3	Y_in_4
1.1330	1.0633	0.8939	0.9908
1.2224	1.1539	0.9798	1.0995
1.1829	1.0567	0.9861	1.1331
0.7581	0.7546	0.6820	0.7104
0.7392	0.7252	0.7106	0.7770
1.0334	0.9418	0.9932	1.0516
0.9321	0.7986	0.8533	0.8993
0.7463	0.7590	0.7843	0.8534
0.6707	0.7574	0.8184	0.8064

f. Menerapkan Fungsi Aktivasi Menghitung Sinyal *Output*

Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner. Berikut adalah contoh manualisasi proses perhitungan sinyal *output* pada data ke-1 :

$$Y_1 = \frac{1}{(1+2.71828)^{-(1.1330)}} = 0.7564$$

$$Y_2 = \frac{1}{(1+2.71828)^{-(1.0633)}} = 0.7433$$

$$Y_3 = \frac{1}{(1+2.71828)^{-(0.8939)}} = 0.7097$$

$$Y_4 = \frac{1}{(1+2.71828)^{-(0.9908)}} = 0.7292$$

Berikut adalah hasil perhitungan sinyal *output* dengan menerapkan fungsi aktivasi sigmoid biner pada Tabel 4.15 Hasil Perhitungan Sinyal *Output* dengan Fungsi Sigmoid Biner sebagai berikut :

Tabel 4.15 Hasil Perhitungan Sinyal *Output* dengan Fungsi Sigmoid biner

Y1	*Y2*	*Y3*	*Y4*
0.7564	0.7433	0.7097	0.7292
0.7725	0.7602	0.7271	0.7502
0.7655	0.7421	0.7283	0.7564
0.6809	0.6802	0.6642	0.6705
0.6768	0.6737	0.6705	0.6850
0.7376	0.7195	0.7297	0.7411
0.7175	0.6897	0.7013	0.7108
0.6784	0.6811	0.6866	0.7013
0.6617	0.6808	0.6939	0.6914

1.2 Perhitungan *Backpropagation*

Proses selanjutnya adalah perhitungan *backpropagation*. Tahapan-tahapan pada proses *backpropagation* adalah sebagai berikut :

a. Menghitung *Error Output*

Pada proses ini, unit *output* berusaha untuk mencapai nilai target yang sesuai dengan pola *input* pelatihan. Setelah itu, menghitung *error output* yang dihasilkan. Berikut adalah target dari data yang telah dinormalisasikan dari data ke-1 pada Tabel 4.16 Target Data ke-1 Sesudah Normalisasi sebagai berikut :

Tabel 4.16 Target Data ke-1 Sesudah Normalisasi

No	Nama Perlakuan	*Output*			
		Protein	Lemak	Laktosa	*Density*
1	Premium	0.8500	0.4835	0.8900	0.8447

Contoh manualisasi perhitungan *error output* dari target data ke-1 sesudah normalisasi adalah sebagai berikut :

$$\delta_1 = (0.85 - 0.7564) * 0.7564 * (1 - 0.7564) = 0.0172$$

$$\delta_2 = (0.4835 - 0.7433) * 0.7433 * (1 - 0.7433) = -0.0496$$

$$\delta_3 = (0.89 - 0.7097) * 0.7097 * (1 - 0.7097) = 0.0371$$

$$\delta_4 = (0.8447 - 0.7292) * 0.7292 * (1 - 0.7292) = 0.0228$$

Berikut adalah hasil *error* antara *output* dengan target yang diberikan sebagai data latih pada Tabel 4.17 Hasil Perhitungan *Error Output* ke Target :

Tabel 4.17 Hasil Perhitungan *Error Output* ke Target

δ_1	δ_2	δ_3	δ_4
0.0172	-0.0496	0.0371	0.0228
0.0004	-0.0587	0.0244	0.0281
-0.1105	0.0302	-0.0195	-0.0538
-0.0393	-0.1020	0.0191	0.0201
0.0488	-0.0146	0.0507	0.0311
-0.0847	-0.1250	-0.1242	-0.1230
-0.1100	-0.0159	-0.0149	-0.0358
-0.1262	-0.0187	-0.0186	-0.0400
-0.1145	-0.0841	-0.0178	-0.0197

b. Menghitung Perubahan Bobot pada *Output Layer*

Langkah selanjutnya adalah proses menghitung perubahan bobot. Proses ini menggunakan nilai *learning rate* = 0.64. Proses menghitung perubahan bobot merupakan proses yang sekaligus mengirimkan informasi *error* yang didapatkan ke lapisan *output*. Berikut adalah contoh manualisasi perhitungan perubahan bobot terhadap *output layer* sebagai berikut :

$$\Delta W_{11} = 0.64 * 0.0172 * 0.4816 = 0.0053$$

$$\Delta W_{12} = 0.64 * -0.0496 * 0.4816 = -0.0153$$

$$\Delta W_{13} = 0.64 * 0.0371 * 0.4816 = 0.0115$$

$$\Delta W_{14} = 0.64 * 0.0228 * 0.4816 = 0.0070$$

Berikut adalah hasil perhitungan perubahan bobot yang didapatkan pada data latih ke-1 di Tabel 4.18 Hasil Perhitungan Perubahan Bobot *Output Layer* :

Tabel 4.18 Hasil Perhitungan Perubahan Bobot *Output Layer*

ΔW_{11}	ΔW_{21}	ΔW_{31}	ΔW_{41}	ΔW_{12}	ΔW_{22}	ΔW_{32}	ΔW_{42}
0.0053	0.0016	0.0005	0.0002	-0.0153	-0.0047	-0.0015	-0.0004
ΔW_{13}	ΔW_{23}	ΔW_{33}	ΔW_{43}	ΔW_{14}	ΔW_{24}	ΔW_{34}	ΔW_{44}
0.0115	0.0035	0.0011	0.0003	0.0070	0.0022	0.0007	0.0002

c. Menjumlahkan Delta *Input*

Setelah itu, setiap unit *hidden layer* melakukan proses penjumlahan setiap delta *input*-nya pada unit-unit yang berada pada *output layer*. Berikut adalah contoh manualisasi perhitungan menjumlahkan delta *input* sebagai berikut :

$$\delta_in_1 = (0.0172 * 0.5716) + (-0.0496 * 0.9383) + (0.0371 * 0.3699) + (0.0228 * 0.5031)$$
$$= -0.0055$$

$$\delta_in_2 = (0.0172 * 0.4962) + (-0.0496 * 0.6603) + (0.0371 * 0.7822) + (0.0228 * 0.3135)$$
$$= 0.0277$$

$$\delta_in_3 = (0.0172 * 0.1278) + (-0.0496 * 0.7164) + (0.0371 * 0.1866) + (0.0228 * 0.8484)$$
$$= -0.0084$$

$$\delta_in_4 = (0.0172 * 0.1989) + (-0.0496 * 0.7722) + (0.0371 * 0.7479) + (0.0228 * 0.3829)$$
$$= 0.0334$$

Berikut adalah hasil menjumlahkan setiap unit *hidden* ke *output* terhadap semua data latih yang telah dinormalisasi pada Tabel 4.19. Hasil Menjumlahkan Delta Unit *Hidden* ke *Output* sebagai berikut :

Tabel 4.19 Hasil Menjumlahkan Delta Unit *Hidden* ke *Output*

δ_in1	*δ_in2*	*δ_in3*	*δ_in4*
-0.0055	0.0277	-0.0084	0.0334
-0.0188	0.0012	-0.0201	0.0133
-0.0654	-0.1422	-0.0639	-0.0858
-0.0607	-0.0700	-0.0708	-0.0232
0.0325	0.0937	0.0358	0.0721
-0.1429	-0.3370	-0.2348	-0.2249
-0.0638	-0.1330	-0.0635	-0.0685
-0.0634	-0.1438	-0.0625	-0.0691
-0.0690	-0.1422	-0.0761	-0.0591

d. Menghitung Informasi *Error*

Berikut adalah contoh manualisasi perhitungan informasi *error* yang didapatkan dari *hidden layer* dengan menggunakan data latih ke-1 pada iterasi pertama sebagai berikut :

$$\delta_1 = -0.0055 * 0.4816 * (1 - 0.4816) = -0.0014$$

$$\delta_2 = 0.0277 * 0.4770 * (1 - 0.4770) = 0.0069$$

$$\delta_3 = -0.0084 * 0.4599 * (1 - 0.4599) = -0.0021$$

$$\delta_4 = 0.0334 * 0.4771 * (1 - 0.4771) = 0.0083$$

Berikut adalah hasil perhitungan informasi *error* pada *hidden layer* untuk semua data latih pada iterasi pertama pada Tabel 4.20 Hasil Menghitung Informasi *Error* sebagai berikut :

Tabel 4.20 Hasil Menghitung Informasi *Error*

δ_1	δ_2	δ_3	δ_4
-0.0014	0.0069	-0.0021	0.0083
-0.0047	0.0003	-0.0050	0.0033
-0.0159	-0.0355	-0.0157	-0.0215
-0.0144	-0.0160	-0.0154	-0.0054
0.0072	0.0220	0.0085	0.0165
-0.0353	-0.0842	-0.0587	-0.0560
-0.0160	-0.0331	-0.0159	-0.0171
-0.0148	-0.0359	-0.0155	-0.0171
-0.0172	-0.0355	-0.0190	-0.0148

e. Perubahan Bobot pada *Hidden Layer*

Tahapan terakhir pada proses *backpropagation* adalah proses menghitung koreksi perubahan bobot pada *hidden layer*. Berikut adalah contoh manualisasi perhitungan perubahan bobot pada *hidden layer* sebagai berikut :

$$\Delta V_{11} = 0.64 * \text{-}0.0014 * 0.6867 = \text{-}0.0006$$

$$\Delta V_{21} = 0.64 * \text{-}0.0014 * 0.4126 = \text{-}0.0004$$

$$\Delta V_{31} = 0.64 * \text{-}0.0014 * 0.3963 = \text{-}0.0003$$

$$\Delta V_{41} = 0.64 * \text{-}0.0014 * 0.3963 = \text{-}0.0003$$

Berikut adalah perubahan bobot pada *hidden layer* yang didapatkan pada data latih ke-1 iterasi pertama pada Tabel 4.21 Hasil Perubahan Bobot *Hidden Layer* sebagai berikut :

Tabel 4.21 Hasil Perubahan Bobot *Hidden Layer*

ΔV_{11}	ΔV_{21}	ΔV_{31}	ΔV_{41}	ΔV_{12}	ΔV_{22}	ΔV_{32}	ΔV_{42}
-0.0006	-0.0004	-0.0003	-0.0003	0.0030	0.0018	0.0018	0.0016
ΔV_{13}	ΔV_{23}	ΔV_{33}	ΔV_{43}	ΔV_{14}	ΔV_{24}	ΔV_{34}	ΔV_{44}
-0.0009	-0.0006	-0.0005	-0.0005	0.0037	0.0022	0.0021	0.0019

f. Hitung MSE

Berikut adalah contoh manualisasi perhitungan mencari nilai *error* dan nilai MSE. Nilai MSE yang didapatkan untuk data ke-1 dan rata-rata *error* yang dihasilkan oleh data latih pada iterasi pertama diperlihatkan pada Tabel 4.22 Hasil Hitung *Error* pada Iterasi Pertama sebagai berikut :

$$error = \frac{1}{4} * ((0.85 - 0.7564) + (0.4835 - 0.7433) + (0.89 - 0.7097) + (0.8447 - 0.7292)) = 0.0324$$

$$MSE = \frac{1}{9} * \sum (error)^2 = 0.0656$$

Tabel 4.22 Hasil Hitung *Error* pada Iterasi Pertama

Data X_i	*Error*
X_1	0.0324
X_2	-0.0117
X_3	-0.2119
X_4	-0.1182
X_5	0.1325
X_6	-0.5820
X_7	-0.2155
X_8	-0.2356
X_9	-0.2687
MSE	0.0656

1.3 Perhitungan Perbaikan Bobot

Pada perhitungan tahap ini akan dicari bobot terbaru pada setiap unit *hidden* maupun unit *output*. Berikut adalah langkah-langkah proses manualisasinya :

1. Perubahan Bobot pada Unit *Hidden*

Contoh proses manualisasi perhitungan bobot pada unit *hidden* untuk data latih ke-2 sebagai berikut :

$$V_{11} = 0.7105 + \text{-}0.0006 = 0.7099$$

$$V_{21} = -0.5918 + \text{-}0.0004 = \text{-}0.5922$$

$$V_{31} = -0.6764 + \text{-}0.0003 = \text{-}0.6767$$

$$V_{41} = -0.1384 + \text{-}0.0003 = \text{-}0.1387$$

Berikut adalah hasil dari perubahan bobot pada unit *hidden* untuk data latih ke-2 iterasi pertama pada Tabel 4.23 Bobot Baru pada Unit *Hidden* :

Tabel 4.23 Bobot Baru pada Unit *Hidden*

V_{11}	V_{21}	V_{31}	V_{41}	V_{12}	V_{22}	V_{32}	V_{42}
0.7099	-0.5922	-0.6767	-0.1387	0.2555	-0.0917	0.1399	-0.7915
V_{13}	V_{23}	V_{33}	V_{43}	V_{14}	V_{24}	V_{34}	V_{44}
-0.0673	-0.3614	0.7021	-0.6893	0.4054	-0.5543	-0.0487	-0.3295

Bobot terakhir dari iterasi pertama akan digunakan sebagai bobot awal pada ietarsi kedua. Berikut ini merupakan bobot terakhir apda iterasi pertama ditunjukkan pada Tabel 4.24.

Tabel 4.24 Bobot Terakhir pada Unit *Hidden* Iterasi pertama

V_{11}	V_{21}	V_{31}	V_{41}	V_{12}	V_{22}	V_{32}	V_{42}
0.6666	-0.6190	-0.7140	-0.1575	0.1703	-0.1409	0.0654	-0.8195
V_{13}	V_{23}	V_{33}	V_{43}	V_{14}	V_{24}	V_{34}	V_{44}
-0.1225	-0.3974	0.6570	-0.7105	0.3570	-0.5806	-0.0892	-0.3414

2. Perubahan Bobot pada Unit *Output*

Contoh proses manualisasi perhitungan bobot pada unit *output* untuk data latih ke-2 sebagai berikut :

$$W_{11} = 0.5716 + 0.0053 = 0.5769$$

$$W_{21} = 0.4962 + -0.0153 = 0.4809$$

$$W_{31} = 0.1278 + 0.0115 = 0.1393$$

$$W_{41} = 0.1989 + 0.0070 = 0.2059$$

Berikut adalah hasil dari perubahan bobot pada unit *output* untuk data latih ke-2 iterasi pertama pada Tabel 4.25 Bobot Baru pada Unit *Output* sebagai berikut:

Tabel 4.25 Bobot Baru pada Unit *Output*

W_{11}	W_{21}	W_{31}	W_{41}	W_{12}	W_{22}	W_{32}	W_{42}
0.5769	0.4809	0.1393	0.2059	0.9399	0.6556	0.7199	0.7744
W_{13}	W_{23}	W_{33}	W_{43}	W_{14}	W_{24}	W_{34}	W_{44}
0.3704	0.7807	0.1877	0.7486	0.5033	0.3131	0.8487	0.3831

2. Iterasi Kedua

Pada iterasi kedua, lakukan langkah-langkah seperti yang dilakukan pada iterasi pertama, mulai dari prosess *feedforward* sampai dengan proses *backpropagation*. Jika pada iterasi pertama, bobot data ke-1 dibangkitkan secara acak, maka pada iterasi kedua, bobot pertama data ke-1 didapatkan dari proses perubahan bobot terakhir pada iterasi pertama. Perubahan bobot *hidden* data ke-1 pada iterasi kedua sudah ditunjukkan pada Tabel 4.25.

Setelah proses iterasi kedua selesai, maka didapatkan *Mean Square Error* (MSE) yang contoh perhitungannya adalah sebagai berikut.

Berikut ini merupakan contoh perhitungan pencarian nilai MSE untuk data ke-1 pada iterasi kedua:

$$Error = 1/4 * ((0.85 - 0.6486) + (0.4835 - 0.6686) + (0.89 - 0.6798) + (0.8447 -$$

0.6778)) $= 0.0984$

Sedangkan nilai MSE pada iterasi pertama dengan 9 data latih pada iterasi kedua adalah sebagai berikut:

$MSE = \frac{1}{9} * \sum error^2 = 0.0520$

Berikut ini merupakan hasil *error* yang dihasilkan setiap data latih pada iterasi pertama ditunjukkan pada Tabel 4.26.

Tabel 4.26 Hasil *Error* Iterasi ke-2

Data X_i	*Error*
X_1	0.0984
X_2	0.0646
X_3	-0.1429
X_4	-0.0679
X_5	0.1765
X_6	-0.5217
X_7	-0.1676
X_8	-0.2017
X_9	-0.2389
MSE	0.0520

4.2.2.2 *Perhitungan Pengujian Data Uji ANN*

Pada manualisasi tahap ini, dilakukan iterasi sebanyak satu kali. Bobot terakhir yang diperoleh dari data ke-9 pada iterasi kedua dijadikan sebagai bobot terbaik yang akan digunakan pada proses pengujian data uji ANN. Proses pengujian dilakukan hanya pada tahapan *feedforward* saja. Data uji yang diproses sudah dinormalisasi terlebih dahulu sesuai pada Tabel 4.7. Berikut adalah proses pengujian data uji ANN :

a. Proses *Input Layer* ke *Hidden Layer*

Bobot yang digunakan untuk proses *input layer* ke *hidden layer* adalah bobot (V_{ij}) yang didapatkan dari data terakhir pada iterasi kedua. Berikut adalah bobot baru untuk data uji ke-1 yang diperlihatkan pada Tabel 4.27 Bobot *Input Layer* ke *Hidden Layer* Data Uji sebagai berikut :

Tabel 4.27 Bobot *Input Layer* ke *Hidden Layer* Data Uji

V_{11}	V_{21}	V_{31}	V_{41}	V_{12}	V_{22}	V_{32}	V_{42}
0.6434	-0.6321	-0.7301	-0.1648	0.1046	-0.1750	0.0148	-0.8332
V_{13}	V_{23}	V_{33}	V_{43}	V_{14}	V_{24}	V_{34}	V_{44}
-0.1582	-0.4206	0.6329	-0.7208	0.3267	-0.5948	-0.1114	-0.3431

b. Menjumlahkan Bobot Sinyal *Input*

Langkah selanjutnya adalah menjumlahkan bobot sinyal *input*. Contoh manualisasi perhitungan bobot sinyal *input* dari data uji ke-1 sebagai berikut :

$$Z_in_1 = (0.6867 * 0.6434) + (0.4126 * -0.6321) + (0.3963 * -0.7301) + (0.3551 * -0.1648) = -0.1669$$

$$Z_in_2 = (0.6867 * 0.1046) + (0.4126 * -0.1750) + (0.3963 * 0.0148) + (0.3551 * -0.8332) = -0.2904$$

$$Z_in_3 = (0.6867 * -0.1582) + (0.4126 * -0.4206) + (0.3963 * 0.6329) + (0.3551 * -0.7208) = -0.2874$$

$$Z_in_4 = (0.6867 * 0.3267) + (0.4126 * -0.5948) + (0.3963 * -0.1114) + (0.3551 * -0.3431) = -0.1871$$

Berikut adalah hasil perhitungan penjumlahan bobot sinyal input yang didapatkan dari 6 data uji yang telah dinormalisasi pada Tabel 4.28 Jumlah Z_in_j Bobot Sinyal Data Uji sebagai berikut :

Tabel 4.28 Jumlah Z_in_j Bobot Sinyal *Input* Data Uji

Z_in_1	Z_in_2	Z_in_3	Z_in_4
-0.1669	-0.2904	-0.2874	-0.1871
-0.1596	-0.2093	0.0236	-0.0227
-0.4369	-0.2115	0.1112	-0.1504
-0.5603	-0.8329	-0.8997	-0.6694
-0.7542	-0.6197	-0.5252	-0.6418
-0.2942	-0.1496	-0.1652	-0.2237

c. Menerapkan Fungsi Aktivasi Menghitung Sinyal *Input*

Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner. Berikut adalah contoh manualisasi proses perhitungan sinyal *input* :

e = bilangan *euler*, yaitu 2.71828

$$Z_1 = \frac{1}{(1 + 2.71828^{-(-0.1669)})} = 0.4584$$

$$Z_2 = \frac{1}{(1 + 2.71828^{-(-0.2904)})} = 0.4279$$

$$Z_3 = \frac{1}{(1 + 2.71828^{-(-0.2874)})} = 0.4286$$

$$Z_4 = \frac{1}{(1 + 2.71828^{-(-0.1871)})} = 0.4534$$

Berikut adalah hasil sinyal *input* dengan menerapkan fungsi aktivasi sigmoid biner pada Tabel 4.29 Hasil Perhitungan Sinyal *Input* Data Uji sebagai berikut :

Tabel 4.29 Hasil Perhitungan Sinyal *Input* Data Uji

Z_1	Z_2	Z_3	Z_4
0.4584	0.4279	0.4286	0.4534
0.4602	0.4479	0.5059	0.4943
0.3925	0.4473	0.5278	0.4625
0.3635	0.3030	0.2891	0.3386
0.3199	0.3498	0.3716	0.3448
0.4270	0.4627	0.4588	0.4443

d. Bobot dari *Hidden Layer* ke *Output Layer*

Berikut adalah bobot dari *hidden layer* ke *output layer* untuk proses pengujian yang diperlihatkan pada Tabel 4.30 Bobot dari *Hidden Layer* ke *Output Layer* Data Uji sebagai berikut :

Tabel 4.30 Bobot dari *Hidden Layer* ke *Output Layer* Data Uji

W_{11}	W_{21}	W_{31}	W_{41}	W_{12}	W_{22}	W_{32}	W_{42}
0.07617	0.12981	0.07519	0.05187	0.57644	0.39795	0.64039	0.61594
W_{13}	W_{23}	W_{33}	W_{43}	W_{14}	W_{24}	W_{34}	W_{44}
0.02296	0.56646	0.11234	0.59823	0.20230	0.11649	0.78240	0.25437

e. Menjumlahkan Setiap Bobot Sinyal *Output*

Langkah selanjutnya adalah menjumlahkan setiap bobot sinyal *output*. Contoh manualisasi perhitungan bobot sinyal *output* sebagai berikut :

$$Y_in_1 = (0.4584 * 0.07617) + (0.4279 * 0.57644) + (0.4286 * 0.02296) + (0.4534 * 0.20230)$$
$$= 0.3831$$

$$Y_in_2 = (0.4584 * 0.12981) + (0.4279 * 0.39795) + (0.4286 * 0.56646) + (0.4534 * 0.11649)$$
$$= 0.5254$$

$$Y_in_3 = (0.4584 * 0.07519) + (0.4279 * 0.64039) + (0.4286 * 0.11234) + (0.4534 * 0.78240)$$
$$= 0.7114$$

$$Y_in_4 = (0.4584 * 0.05187) + (0.4279 * 0.61594) + (0.4286 * 0.59823) + (0.4534 * 0.25437)$$
$$= 0.6591$$

Berikut adalah hasil menjumlahkan bobot sinyal *output* yang didapatkan dari 6 data uji pada Tabel 4.31 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer* Data Uji sebagai berikut :

Tabel 4.31 Jumlah Y_in_k Bobot Sinyal *Hidden Layer* ke *Output Layer* Data Uji

Y_in_1	Y_in_2	Y_in_3	Y_in_4

0.3831	0.5254	0.7114	0.6591
0.4048	0.5821	0.7650	0.7281
0.3934	0.5818	0.7371	0.7293
0.2775	0.3710	0.5188	0.4646
0.3043	0.4314	0.5596	0.5421
0.3997	0.5512	0.7276	0.6946

f. Menerapkan Fungsi Aktivasi Menghitung Sinyal *Output*

Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid biner. Berikut adalah contoh manualisasi proses perhitungan sinyal *output* sebagai berikut :

$$Y_1 = \frac{1}{(1+2.71828^{-(0.3831)})} = 0.5946$$

$$Y_2 = \frac{1}{(1+2.71828^{-(0.5254)})} = 0.6284$$

$$Y_3 = \frac{1}{(1+2.71828^{-(0.7114)})} = 0.6707$$

$$Y_4 = \frac{1}{(1+2.71828^{-(0.6591)})} = 0.6591$$

Berikut adalah hasil perhitungan sinyal *output* dengan menerapkan fungsi aktivasi sigmoid biner pada Tabel 4.32 Hasil Perhitungan Sinyal *Output* Data Uji sebagai berikut :

Tabel 4.32 Hasil Perhitungan Sinyal *Output* Data Uji

Y1	*Y2*	*Y3*	*Y4*
0.5946	0.6284	0.6707	0.6591
0.5999	0.6416	0.6824	0.6744
0.5971	0.6415	0.6764	0.6746
0.5689	0.5917	0.6269	0.6141
0.5755	0.6062	0.6364	0.6323
0.5986	0.6344	0.6743	0.6670

4.2.2.3 *Proses Denormalisasi*

Proses denormalisasi adalah proses yang digunakan untuk mengubah nilai kandungan gizi susu ke dalam *range* nilai aslinya. Y_1 adalah *output* dari kandungan gizi susu berupa protein, Y_2 adalah *output* dari kandungan gizi susu berupa lemak, Y_3 adalah *output* dari kandungan gizi susu berupa laktosa dan Y_4 adalah *output* dari kandungan gizi susu berupa *density*. Berikut adalah proses denormalisasi pada Tabel 4.33 Hasil Denormalisasi Data Uji sebagai berikut :

$$Y_1'' = (((4.13 - 3.81) * (0.5946 - 0.1))/0.8) + 3.81 = 4.0079$$

$$Y_2'' = (((8.19 - 6.25) * (0.6284 - 0.1))/0.8) + 6.25 = 7.5314$$

$$Y_3'' = (((3.94 - 3.14) * (0.6707 - 0.1)) / 0.8) + 3.14 = 3.7107$$

$$Y_4'' = (((28.95 - 22.3) * (0.6591 - 0.1)) / 0.8) + 22.3 = 26.9471$$

Tabel 4.33 Hasil Denormalisasi Data Uji

Y_1''	Y_2''	Y_3''	Y_4''
4.0079	7.5314	3.7107	26.9471
4.0099	7.5633	3.7224	27.0746
4.0088	7.5631	3.7164	27.0767
3.9976	7.4424	3.6669	26.5735
4.0002	7.4776	3.6764	26.7248
4.0094	7.5460	3.7143	27.0131

4.2.3 Proses Perhitungan Manual *Particle Swarm Optimization* (PSO)

Pada contoh perhitungan manual ini, *user* memilih jenis rumput yang akan dimodelkan, yaitu rumput Raja, rumput Taiwan atau rumput Odot. Sedangkan hasil nya nanti merupakan kandungan gizi susu yang dihasilkan dari *input*-an kandungan nutrisi rumput dari *user*.

Jumlah dimensi yang dibentuk oleh sistem sebesar 4 yang merupakan kandungan nutrisi dari rumput yang dipilih *user*. Selain dimensi permasalahan juga terdapat beberapa parameter lain yang telah ditentukan sebelumnya dalam algoritma PSO dan inisialisasi parameter adalah sebagai berikut :

a. *Swarm size* = 9
b. *Problem dimension* = 4
c. c_1 = 2
d. c_2 = 2
e. w_{min} = 0,4
f. w_{max} = 0,9
g. *Max. iteration*(t) = 3

4.2.3.1 *Inisialisasi partikel*

Setelah menentukan parameter algoritma PSO, langkah pertama adalah inisialisasi partikel. *Range* atau batas bawah dan batas atas partikel pada kasus ini menggunakan kandungan nutrisi dari jenis rumput yaitu rumput Odot dan rumput Raja. Nilai setiap kandungan nutrisi pada Tabel 4.34.

Tabel 4.34 *Range* Minimal dan Maksimal

Range	Kandungan Nutrisi Odot			
	Abu	PK	LK	SK
Min	1.184	0.989	0.171	1.653
Max	1.334	1.163	0.198	1.907

Selanjutnya inisialisasi partikel dibentuk sesuai dengan persamaan 2.19. Berikut merupakan contoh perhitungannya, misalkan untuk membentuk partikel x_1 pada dimensi 0 yaitu abu.

$x_i(0) = x_{min,d} + r_d(x_{max,d} - x_{min,d}) = 1.184 + rand(0,1) * (1.334 - 1.184) = 1.299$

Dimana x_i merupakan partikel ke-i , d adalah dimensi dan t adalah iterasi. Nilai *random* di-*generate* dalam rentang [0,1] dan hasil inisialiasi seluruh partikel untuk setiap dimensi ditunjukkan pada Tabel 4.35. Nilai kecepatan awal partikel ada 0 dan ditunjukkan pada Tabel 4.36.

Tabel 4.35 Inisialisasi partikel PSO

$x(i)$	t	*d*			
		0	1	2	3
x1	0	1.299	1.109	0.173	1.902
x2	0	1.267	1.045	0.181	1.897
x3	0	1.194	1.018	0.193	1.787
x4	0	1.290	1.058	0.195	1.877
x5	0	1.207	1.071	0.180	1.716
x6	0	1.280	1.110	0.191	1.876
x7	0	1.271	1.017	0.173	1.731
x8	0	1.252	1.126	0.183	1.751
x9	0	1.328	1.152	0.196	1.658

Tabel 4.36 Kecepatan Awal Partikel

V iterasi 0					
$x(i)$	t	*d*			
		0	1	2	3
x1	0	0	0	0	0
x2	0	0	0	0	0
x3	0	0	0	0	0
x4	0	0	0	0	0
x5	0	0	0	0	0
x6	0	0	0	0	0
x7	0	0	0	0	0
x8	0	0	0	0	0
x9	0	0	0	0	0

4.2.3.2 *Menghitung fungsi objektif PSO*

Proses berikutnya adalah menghitung fungsi obyektif atau nilai fitness untuk setiap partikel dengan menggunakan Persamaan 2.24. Dalam proses pencarian nilai fitness, setiap partikel akan melalui proses *feedforward* terlebih dahulu sampai didapatkan nilai output dari *feedforward*. Berikut contoh prosesnya:

1. **Proses *Input Layer* ke *Hidden Layer***

Proses *input layer* ke *hidden layer* V_{ij} bobot dibangkitkan berdasarkan bobot terakhhir pada proses pengujian ANN, yang ditunjukkan pada Tabel 4.37.

Tabel 4.37 Bobot *input* ke *hidden layer*

V_{ij}	Z_1	Z_2	Z_3	Z_4
X_1	0.660	0.093	-0.137	0.318
X_2	-0.630	-0.209	-0.417	-0.613
X_3	-0.722	-0.019	0.635	-0.131
X_4	-0.166	-0.873	-0.724	-0.367

2. **Menjumlahkan bobot sinyal *input***

Berikut cara menjumlahkan bobot sinyal *input*:

$Z_in_1 =$ (1.299 *0.660) + (1.109 *-0.630) + (0.173 *-0.722) + (1.902 *-0.166)

$=$ -0.2822

$Z_in_2 =$ (1.299 *0.093) + (1.109 *--0.209) + (0.173 *--0.019) + (1.902 *-0.873)

$=$ -1.4699

$Z_in_3 =$ (1.299 *-0.137) + (1.109 *-0.417) + (0.173 *0.635) + (1.902 *0.724)

$=$ -1.5027

$Z_in_4 =$ (1.299 *0.318) + (1.109 *-0.613) + (0.173 *-0.131) + (1.902 *0.367)

$=$ -0.1835

Berikut ini merupakan hasil dari menjumlahkan bobot sinyal *input* dari data latih yang normalisasi seperti pada Tabel 4.38.

Tabel 4.38 Jumlah bobot sinyal *input*

Z_in_1	Z_in_2	Z_in_3	Z_in_4
-0.2822	-1.4699	-1.5027	-0.1835
-0.2680	-1.4054	-1.4672	-0.1858
-0.2895	-1.3522	-1.3860	-0.1766
-0.2678	-1.4334	-1.4805	-0.1909
-0.2932	-1.4031	-1.3905	-0.1698
-0.3041	-1.4641	-1.4821	-0.1835
-0.2144	-1.4019	-1.4316	-0.1843

-0.3061	-1.4684	-1.4392	-0.1734
-0.2663	-1.5459	-1.4777	-0.1845

3. **Menerapkan Fungsi Aktivasi dalam Menghitung Sinyal *Input***

Fungsi aktivasi yang digunakan adalah *sigmoid biner*, berikut merupakan perhitungan dalam mencari sinyal *input.*

e = Bilangan *euler* = 2.71828

$$Z_1 = \frac{1}{(1+2.71828)^{-(-0.2822)}} = 0.4299$$

$$Z_2 = \frac{1}{(1+2.71828)^{-(-1.4699)}} = 0.1870$$

$$Z_3 = \frac{1}{(1+2.71828)^{-(-1.5027)}} = 0.1820$$

$$Z_4 = \frac{1}{(1+2.71828)^{-(-0.1835)}} = 0.4542$$

Berikut ini hasil dari sinyal *input* yang menerapkan fungsi aktivasi *sigmoid biner*, ditunjukkan pada Tabel 4.39.

Tabel 4.39 Hasil Perhitungan sinyal *Input* dengan Aktivasi *Sigmoid Biner*

Z_1	Z_2	Z_3	Z_4
0.4299	0.1870	0.1820	0.4542
0.4334	0.1970	0.1874	0.4537
0.4281	0.2055	0.2001	0.4560
0.4334	0.1926	0.1854	0.4524
0.4272	0.1973	0.1993	0.4577
0.4246	0.1878	0.1851	0.4542
0.4466	0.1975	0.1929	0.4541
0.4241	0.1872	0.1917	0.4568
0.4338	0.1757	0.1858	0.4540

4. **Menghitung Bobot dari *Hidden Layer* ke *Output Layer***

Data untuk bobot didapatkan dari bobot terakhir pada pengujian ANN. Berikut ini merupakan bobot dari *hidden layer* ke *output layer,* ditunjukkan pada Tabel 4.40.

Tabel 4.40 Bobot dari *Hidden Layer* ke *Output Layer*

W_{jk}	Y_1	Y_2	Y_3	Y_4
Z_1	-0.3066	0.3069	-0.1338	0.1137
Z_2	0.2716	0.4954	0.6188	0.1496

Z_3	-0.0293	0.5898	0.0842	0.7670
Z_4	-0.1134	0.5319	0.5510	0.2271

5. **Menjumlahkan setiap bobot sinyal *output.***

$Y_{in_1} = (0.4299 * -0.3066) + (0.1870 * 0.3069) + (0.1820 * -0.1338) +$ (0.4542*0.1137)= -0.0471

$Y_{in_2} = (0.4299 * 0.2716) + (0.1870 * 0.4954) + (0.1820 * 0.6188) +$ (0.4542*0.1496)= 0.3900

$Y_{in_3} = (0.4299 * -0.0293) + (0.1870 * 0.5898) + (0.1820 * 0.0842) +$ (0.4542*0.7670)= 0.4614

$Y_{in_4} = (0.4299 * -0.1134) + (0.1870 * 0.5319) + (0.1820 * 0.5510) +$ (0.4542*0.2271)= 0.2541

Berikut ini hasil dari sinyal *output*, ditunjukkan pada Tabel 4.41.

Tabel 4.41 Sinyal *output*

Y_in_1	Y_in_2	Y_in_3	Y_in_4
-0.0471	0.3900	0.4614	0.2541
-0.0459	0.3991	0.4672	0.2619
-0.0431	0.4101	0.4752	0.2745
-0.0471	0.3955	0.4635	0.2581
-0.0451	0.4056	0.4717	0.2702
-0.0456	0.3909	0.4623	0.2569
-0.0505	0.4064	0.4679	0.2638
-0.0463	0.3948	0.4645	0.2608
-0.0523	0.3877	0.4548	0.2497

6. **Menerapkan Fungsi Aktivasi dalam Menghitung Sinyal *Output***

Fungsi aktivasi yang digunakan adalah *sigmoid biner*, berikut merupakan perhitungan dalam mencari sinyal *output.*

$$Y_1 = \frac{1}{(1+2.71828)^{-(-0.0471)}} = 0.4882$$

$$Y_2 = \frac{1}{(1+2.71828)^{-(0.3900)}} = 0.5963$$

$$Y_3 = \frac{1}{(1+2.71828)^{-(0.4614)}} = 0.6133$$

$$Y_4 = \frac{1}{(1+2.71828)^{-(0.2541)}} = 0.5632$$

Berikut ini hasil dari sinyal *output* yang menerapkan fungsi aktivasi *sigmoid biner*, ditunjukkan pada Tabel 4.42.

Tabel 4.42 Hasil Sinyal *Output* dengan Fungsi Aktivasi *Sigmoid Biner*

Y_1	Y_2	Y_3	Y_4
0.4882	0.5963	0.6133	0.5632
0.4885	0.5985	0.6147	0.5651
0.4892	0.6011	0.6166	0.5682
0.4882	0.5976	0.6138	0.5642
0.4887	0.6000	0.6158	0.5672
0.4886	0.5965	0.6136	0.5639
0.4874	0.6002	0.6149	0.5656
0.4884	0.5974	0.6141	0.5648
0.4869	0.5957	0.6118	0.5621

Proses Denormalisasi

Proses denormalisasi dilakukan untuk mengubah kembali nilai kandungan gizi susu ke dalam nilai *real* atau aslinya. Berikut adalah contoh perhitungan proses denormalisasi sesuai persamaan 2.17, nilai *min max* data berdasarkan Tabel 4.34 dan hasil yang ditunjukkan pada Tabel 4.43.

Y_1"= ((8.7-3.79)*(0.4882-0.1))/0.8+3.79=6.1727

Y_2"= ((13.59-6.25)*(0.5963 -0.1))/0.8+6.25=10.8033

Y_3"= ((8.29-3.14)*(0.6133 -0.1))/0.8+3.14=6.4447

Y_4"= ((61.45-22.3)*(0.5632 -0.1))/0.8+22.3=44.9673

Tabel 4.43 Hasil Denormalisasi Output

Protein	Lemak	Laktosa	Berat Jenis
6.1727	10.8033	6.4447	44.9673
6.1746	10.8235	6.4536	45.0606
6.1789	10.8477	6.4658	45.2126
6.1727	10.8155	6.4478	45.0155
6.1759	10.8378	6.4603	45.1613
6.1750	10.8053	6.4461	45.0008
6.1676	10.8396	6.4546	45.0835
6.1740	10.8141	6.4493	45.0477
6.1647	10.7984	6.4345	44.9140

Dari hasil denormalisasi output, dapat dihitung nilai *Fitness* masing-masing partikel. Dalam kasus ini adalah maksimal optimasi untuk Protein optimal, Lemak

minimal sesuai dengan persamaan 2.23. Berikut ini contoh perhitungan *fitness*, dan hasilnya ditunjukkan pada Tabel 4.44.

$$fitness(i) = optimal + \frac{1}{minimal}$$

$$fitness(i) = protein + \frac{1}{lemak}$$

$$fitness\ (i) = 6.1727 + (1/10.8033) = 6.2653$$

Tabel 4.44 Fungsi objektif PSO pada inisialisasi awal partikel

$x(i)$	t	$F(x)$ fitness
x1	0	6.2653
x2	0	6.2670
x3	0	6.2711
x4	0	6.2651
x5	0	6.2682
x6	0	6.2675
x7	0	6.2598
x8	0	6.2665
x9	0	6.2574

4.2.3.3 *Nilai pBest dan gBest untuk partikel awal*

Posisi terbaik individu ke-i $(pBest_i)$ awal untuk setiap partikel pada waktu $t = 0$ adalah posisi partikel itu sendiri untuk setiap dimensinya sesuai dengan Persamaan 2.21, nilai pBest untuk partikel awal ditunjukkan pada Tabel 4.45 dan hasil gBest yang ditunjukkan pada Tabel 4.46.

Tabel 4.45 pBest untuk inisialisasi partikel

Pbest						
$x\ (i)$	t	*d*				*Fitness*
		0	1	2	3	
x1	0	1.299	1.109	0.173	1.902	6.2653
x2	0	1.267	1.045	0.181	1.897	6.2670
x3	0	1.194	1.018	0.193	1.787	6.2711
x4	0	1.290	1.058	0.195	1.877	6.2651
x5	0	1.207	1.071	0.180	1.716	6.2682
x6	0	1.280	1.110	0.191	1.876	6.2675

x7	0	1.271	1.017	0.173	1.731	6.2598
x8	0	1.252	1.126	0.183	1.751	6.2665
x9	0	1.328	1.152	0.196	1.658	6.2574

Untuk nilai *gBest* dalam kasus maksimal optimasi diperoleh dengan cara mencari nilai maksimal dari *pbest*. Berikut contoh perhitungan pencarian nilai *gbest* dalam kasus maksimal optimasi.

$$f(gBest(0)) = Max\{f(pBest_i(0))\}$$
$$= Max\begin{pmatrix}6.2653\ ;6.2670\ ;\ 6.2711;6.2651;\ 6.2682;6.2675\ ;\\ 6.2598\ ;\ 6.2665\ ;\ 6.2574\end{pmatrix}$$
$$= 6.2711$$

Tabel 4.46 gBest untuk inisialisasi partikel

Gbest						
$x\ (i)$	t	***d***				***Fitness***
		0	**1**	**2**	**3**	
X7	0	1.285	1.029	0.185	1.903	6.2711

4.2.3.4 *Memperbarui kecepatan*

Proses berikutnya adalah membangkitkan nilai bobot inersia menggunakan Persamaan 2.31 dengan iterasi sebanyak 3 kali. Contoh perhitungan bobot inersia untuk t = 1 adalah sebgai berikut, serta hasil perhitungannya ditunjukkan pada Tabel 4.47.

$$w = (w_{max} - w_{min}) * \frac{(t_{max}-t)}{t_{max}} + w_{min} = (0{,}9 - 0{,}4) * \frac{(3-1)}{3} + 0{,}4 = 0{,}733$$

Tabel 4.47 Nilai bobot inersia

Iterasi (t)	***w***
1	0.73333333
2	0.56666667
3	0.4

Dan untuk pembatasan kecepatan v_{max} diperoleh dengan Persamaan 2.29 dengan nilai *k* dalam rentang $0{,}1 \leq k \leq 1{,}0$ dan *range* sesuai dengan Tabel 4.3. Dibawah ini merupakan contoh perhitungan V_{max} serta hasil yang ditunjukkan pada Tabel 4.48.

$$v_{max} = k * (x_{max} - x_{min}) = 0{,}6 * (1.334 - 1.184) = 0.09$$

Tabel 4.48 Nilai pembatasan kecepatan maksimum setiap dimensi

d	0	1	2	3
v_{max}	0.09	0.1044	0.0162	0.1524

Pembaruan kecepatan v_{id}^{t+1} menggunakan Persamaan 2.21, contoh perhitungan untuk x_1 , d_1 , dan $t = 1$ dan men-generate nilai *random* $r1 = 0{,}01$ dan $r2 = 0{,}17$ sebagai berikut:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * (pBest_i^t - x_{id}^t) + c_2 * r_2 * (gBest^t - x_{id}^t)$$

$$v_{11}^{1} = 0{,}733 * 0 + 2 * 0{,}01 * (1.224 - 1.224) + 2 * 0{,}17 * (6.2711 - 1.224) = 1.6903$$

Jika nilai kecepatan baru lebih dari v_{max}, maka kecepatan baru = v_{max}.

Maka hasil Kecepatan baru pada iterasi pertama ditunjukkan pada Tabel 4.49.

Tabel 4.49 Kecepatan baru pada iterasi pertama

Vbaru iterasi 1					
$x(i)$	*t*	*d*			
		0	1	2	3
x1	1	1.6903	1.7552	2.0735	1.4813
x2	1	1.6999	1.7768	2.0705	1.4844
x3	1	1.7266	1.7884	2.0664	1.5238
x4	1	1.6935	1.7716	2.0655	1.4896
x5	1	1.7223	1.7698	2.0708	1.5502
x6	1	1.6975	1.7539	2.0672	1.4908
x7	1	1.6990	1.7881	2.0737	1.5437
x8	1	1.7075	1.7479	2.0698	1.5380
x9	1	1.6803	1.7380	2.0654	1.5703

4.2.3.5 *Memperbarui posisi*

Setelah mendapatkan nilai kecepatan baru, maka setiap partikel akan bergerak dan memperbaiki posisinya. Posisi baru didapatkan dengan menggunakan Persamaan 2.23 dengan memperhatikan batasan range setiap dimensi pada Tabel 4.43 , jika perubahan posisi melampaui batas maksimum dimensi, maka posisi baru sama dengan batas maksimum dimensi dan jika perubahan posisi mundur melampaui batas minimum dimensi, maka posisi baru sama dengan batas minimum dimensi. Contoh untuk perhitungan posisi baru untuk x_1 , d_1 , dan $t = 1$ serta hasil posisi baru pada itersi pertama yang ditunjukkan pada Tabel 4.50

$$x_{id}^{t+1} = x_{id}^{t} + V_{id}^{t+1}$$

$$x_{11}^{1} = 1.244 + (1.6903) = 1.3890$$

Tabel 4.50 Posisi baru pada iterasi pertama

Posisi baru iterasi 1

$x\,(i)$	t	d			
		0	1	2	3
x1	1	1.3890	1.2134	0.1892	2.0544
x2	1	1.3570	1.1494	0.1972	2.0494
x3	1	1.2840	1.1224	0.2092	1.9394
x4	1	1.3800	1.1624	0.2112	2.0294
x5	1	1.2970	1.1754	0.1962	1.8684
x6	1	1.3700	1.2144	0.2072	2.0284
x7	1	1.3610	1.1214	0.1892	1.8834
x8	1	1.3420	1.2304	0.1992	1.9034
x9	1	1.4180	1.2564	0.2122	1.8104

Setelah mendapatkan posisi baru seluruh partikel maka dilanjutkan dengan mencari nilai fitness yang baru. Nilai fitness dihasilkan melalui proses *feedforward* seperti pada subBab 4.2.3.2. Fungsi objektif pada iterasi pertama ditunjukkan pada Tabel 4.51.

Tabel 4.51 Fungsi objektif PSO pada iterasi pertama

$x(i)$	t	$F(x)$ *fitness*
x1	1	6.2512
x2	1	6.2511
x3	1	6.2523
x4	1	6.2516
x5	1	6.2540
x6	1	6.2517
x7	1	6.2500
x8	1	6.2517
x9	1	6.2537

4.2.3.6 *Memperbarui pBest dan gBest*

Untuk kasus maksimalisasi, nilai *pbest* = fitness jika $f\left(x_{id}(t+1)\right) \leq f\left(pBest(0)\right)$. Sebagai contoh untuk mencari posisi terbaik partikel-1(x_1) pada iterasi ke-1 yaitu dengan membandingkan $f\left(x_{id}(1)\right)$ dengan $f\left(pBest(0)\right)$

$$pBest_i(t+1) = \begin{cases} pBest_i(t) \;\; jika\, f(x_{id}(t+1)) \leq f(pBest_i(t)) \\ x_{id}(t+1) \;\; jika\, f\left(x_{id}(t+1)\right) > f(pBest_i(t)) \end{cases}$$

$f\left(x_{id}(1)\right) = 6.2512$ dan $f\left(pBest(0)\right)$ =6.2653

Karena $6.2512 < 6.2653$ sehingga $f(x_{id}(1)) < f(pBest(0))$, maka posisi terbaik individu adalah posisi pada iterasi ke-0 $f(pBest(0))$. Sehingga hasil *pBest* iterasi pertama ditunjukkan pada Tabel 4.52 dan hasil *gBest* pada Tabel 4.53.

Tabel 4.52 pBest pada iterasi pertama

Pbest						
$x(i)$	t	*d*				*Fitness*
		0	1	2	3	
x1	1	1.3890	1.2134	0.1892	2.0544	6.2653
x2	1	1.3570	1.1494	0.1972	2.0494	6.2670
x3	1	1.2840	1.1224	0.2092	1.9394	6.2711
x4	1	1.3800	1.1624	0.2112	2.0294	6.2651
x5	1	1.2970	1.1754	0.1962	1.8684	6.2682
x6	1	1.3700	1.2144	0.2072	2.0284	6.2675
x7	1	1.3610	1.1214	0.1892	1.8834	6.2598
x8	1	1.3420	1.2304	0.1992	1.9034	6.2665
x9	1	1.4180	1.2564	0.2122	1.8104	6.2574

Tabel 4.53 *gBest* pada iterasi pertama

Gbest						
$x(i)$	t	*d*				*Fitness*
		0	1	2	3	
X3	1	1.2840	1.1224	0.2092	1.9394	6.2711

4.2.3.7 *Random Injection*

Pada kasus ini ditentukan jumlah partikel yang diinjeksi adalah 30% partikel dari ukuran *swarmsize* serta interval *injection* adalah setiap kelipatan 3 iterasi. *Random injection* dilakukan dengan mengevaluasi nilai fitness partikel saat memasuki interval *inject*. Tiga partikel dengan nilai fitness terkecil akan dibuang dan digantikan dengan partikel baru. Misalkan iterasi sudah memasuki iterasi=3, maka perhitungan ditunjukkan pada Tabel 4.54 dan posisi *random injection* pada Tabel 4.55 sebagai berikut:

Tabel 4.54 Posisi baru di iterasi 3

Posisi						
$x(i)$	t	*d*				*Fitness*
		0	1	2	3	
x1	1	1.3340	1.1630	0.1980	1.9070	6.2517
x2	1	1.3340	1.1630	0.1980	1.9070	6.2517
x3	1	1.3340	1.1630	0.1980	1.9070	6.2517
x4	1	1.3340	1.1630	0.1980	1.9070	6.2517
x5	1	1.3340	1.1630	0.1980	1.9070	6.2517

x6	1	1.3340	1.1630	0.1980	1.9070	6.2517
x7	1	1.3340	1.1630	0.1980	1.9070	6.2517
x8	1	1.3340	1.1630	0.1980	1.9070	6.2517
x9	1	1.3340	1.1630	0.1980	1.9070	6.2517

Tabel 4.55 Posisi baru setelah *random injection*

Posisi						
$x(i)$	t	d				*Fitness*
		0	1	2	3	
x1	**1**	**1.2153**	**1.1431**	**0.1940**	**1.6886**	**6.2695**
x2	**1**	**1.2433**	**1.0636**	**0.1861**	**1.6830**	**6.2634**
x3	**1**	**1.1978**	**1.0436**	**0.1949**	**1.7835**	**6.2714**
x4	1	1.3340	1.1630	0.1980	1.9070	6.2517
x5	1	1.3340	1.1630	0.1980	1.9070	6.2517
x6	1	1.3340	1.1630	0.1980	1.9070	6.2517
x7	1	1.3340	1.1630	0.1980	1.9070	6.2517
x8	1	1.3340	1.1630	0.1980	1.9070	6.2517
x9	1	1.3340	1.1630	0.1980	1.9070	6.2517

4.2.3.8 *Keadaan berhenti PSO*

Iterasi dilanjutkan dengan memperbarui kecepatan, memperbarui posisi, melakukan *feedforward,* menghitung nilai *fitness* kemudian memperbarui *pBest* dan *gBest*. Iterasi berhenti sampai pada dengan maksimum iterasi yaitu $t = 3$. Berikut ini merupakan hasil gBest pada Tabel 4.56

Tabel 4.56 gBest pada iterasi ke-3

Gbest						
$x(i)$	t	d				*Fitness*
		0	1	2	3	
X3	3	1.1978	1.0436	0.1949	1.7835	6.2714

4.2.3.9 *Rekomendasi Pakan*

Berdasarkan *gbest* terakhir didapatkan bahwa partikel ke-3 merupakan individu terbaik yang ditunjukkan pada Tabel 4.56 dengan fitness sebesar 6.2714. Diketahui kombinasi prosentase pemberian pakan yaitu 70% rumput Odot dan 30% rumput Raja. Total kandungan rumput Odot dan rumput Raja ditunjukkan pada Tabel 4.2 kemudian dihitung jumlah kandungan nutrisi pakan seperti pada perhitungan berikut.

$$Jumlah\ Total\ Kandungan_{Odot-Raja} = Kand_{Abu} + Kand_{PK} + Kand_{SK} + Kand_{LK}$$

$$Jumlah\ Total\ Kandungan_{Odot-Raja} = 0.3557 + 0.3053 + 0.0526 + 0.4968 \quad = 1.2104$$

Sedangkan untuk total kandungan nutrisi pakan dari individu terbaik yang diperoleh dari proses ANN dan PSO adalah 4.2198. Berikut perhitungan dalam mencari jumlah total kandungan nutrisi dari individu terbaik berdasarkan Tabel 4.56.

$$Jumlah\ Total\ Kandungan_{ANN-PSO} = 1.1978 + 1.0436 + 0.1949 + 1.7835 = 4.2198$$

Jumlah total kandungan nutrisi rumput akan digunakan dalam mencari jumlah (kg) pakan yang akan diberikan kepada kambing dengan berat badan 32 kg. Berikut contoh perhitungan rekomendasi pakan (kg).

$$Total\ Pemberian\ Pakan_{Odot-Raja}(kg) = \frac{4.2198}{1.2104} = 3.486$$

$$Rekomendasi\ Pakan\ Rumput\ Odot\ (kg) = \frac{70}{100} * 3.486 = 2.440$$

$$Rekomendasi\ Pakan\ Rumput\ Raja\ (kg) = \frac{30}{100} * 3.486 = 1.046$$

Dari hasil tersebut dapat direkomendasikan kepada *user* untuk kambing dengan berat badan 32 kg serta jenis pakan rumput Odot 70% dan rumput Raja 30% diberikan jumlah pakan rumput Odot sebesar 2.44 kg serta rumput Raja sebesar 1.046 kg agar dapat memperoleh hasil kandungan gizi susu yang optimal.

4.3 Perancangan Antarmuka

Pada sub bab ini menjelaskan tentang pembuatan rancangan antar muka. Rancangan antar muka dibuat dengan tujuan untuk menggambarkan keadaan sebenarnya dari implementasi sistem pemodelan komposisi pakan kambing PE untuk optimasi kandungan gizi susu menggunakan ANN dan PSO yang akan dibangun. Pada perancangan antar muka pemodelan komposisi pakan kambing PE untuk optimasi kandungan gizi susu menggunakan ANN dan PSO terdiri dari 2 halaman, yaitu Proses Optimasi dan Kesimpulan

4.3.1 Proses Optimasi

Proses optimasi merupakan proses utama dengan memasukkan jenis pakan, prosentase pakan, parameter ANN, parameter PSO dan kandungan gizi yang dioptimalkan. Berikut keterangan rancangan antar muka proses optimasi yang diperlihatkan pada Gambar 4.17 Rancangan Antar Muka Proses Optimasi :

1. *Window title* yang berupa NIM dan nama pembuat program.
2. Judul program.
3. *Tab control* untuk melakukan proses Optimasi.
4. *Check Box* untuk memilih jenis Pakan.
5. *Radio Button* untuk memilih prosentase pakan
6. *Group box* untuk memasukan nilai parameter PSO. Nilai parameter yang diisikan pada proses optimasi PSO adalah :

   a. Jumlah Partikel

b. Max iterasi

c. C_1

d. C_2

e. W_{max}

f. W_{min}

g. Koefisien *K*

7. *Radio Button* untuk memilih kualitas susu
8. *Button RESET* untuk mengulang parameter PSO dan ANN dan *DEFAULT* untuk mengisi nilai default.
9. *Group box* untuk memasukan nilai parameter ANN. Nilai parameter yang diisikan pada proses optimasi ANN adalah :

a. *Learning Rate*

b. Jumlah hidden layer

c. Jumlah data latih

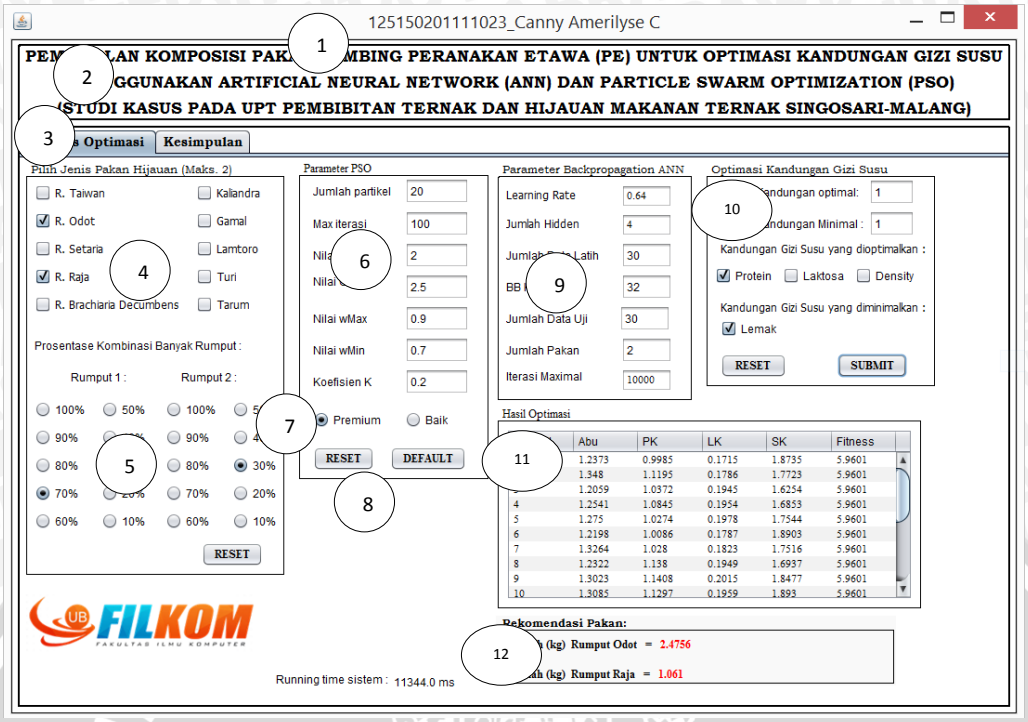
d. BB kambing

e. Jumlah data uji

f. Jumlah Pakan

g. Iterasi Maksimal ANN

10. *Radio button* untuk memilih optimasi kandungan gizi susu
11. Tabel hasil optimasi
12. *Text Field* untuk hasil jumlah pakan (kg)

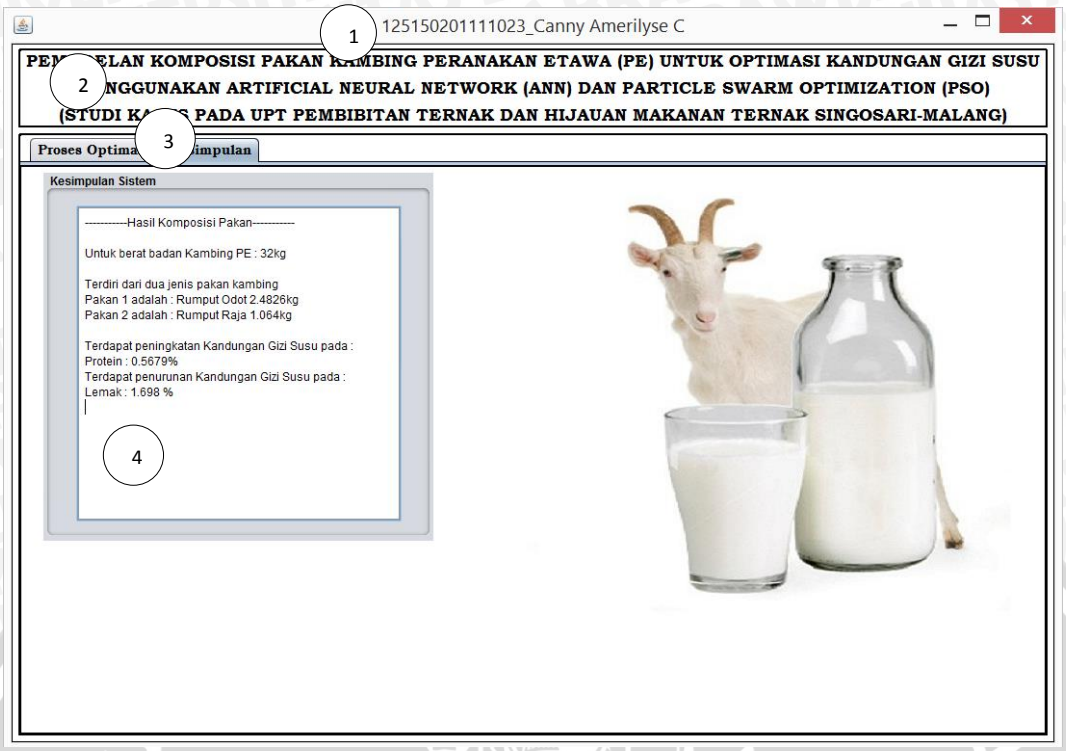


Gambar 4.18 Rancangan Antar Muka Proses Optimasi

4.3.2 Halaman Kesimpulan

Setelah dilakukan proses optimasi, maka akan diberikan kesimpulan yang dihasilkan oleh sistem sebagai rekomendasi hasil yang diberikan. Berikut keterangan rancangan antar muka halaman Kesimpulan yang diperlihatkan pada Gambar 4.18 Rancangan Antar halaman Kesimpulan :

1. *Window title* yang berupa NIM dan nama pembuat program.
2. Judul program.
3. *Tab control* untuk melakukan proses Optimasi.
4. *Text Area* hasil kesimpulan



Gambar 4.19 Rancangan Antar Muka Hasil kesmpulan

4.4 Perancangan Pengujian

Perancangan pengujian akan dilakukan dengan menggunakan data latih dan data uji dari dua jenis kualitas susu, yaitu Kualitas susu Premium dan kualitas Baik. Pada hasil pengujian akan ditampilkan pengujian dari kualitas Premium, sedangkan kualitas Baik akan dilampirkan pada Lampiran.

4.4.1 Pengujian terhadap maksimal iterasi pada data latih ANN

Pengujian iterasi maksimal pada data latih ANN menggunakan dua jenis data latih, yaitu data latih kualitas susu Premium dan kualitas susu Baik. Pengujian ini juga menghasilkan nilai MSE. Hasil pengujian ditunjukkan pada Tabel 4.54.

Tabel 4.57 Pengujian maksimal iterasi data latih ANN

Jumlah Iterasi	**Nilai fitness pada percobaan ke-*i***					**Rata-rata MSE**
	1	**2**	**3**	**4**	**5**	
10						
20						
50						
100						
150						

200						

4.4.2 Pengujian terhadap kombinasi nilai bobot inersia minimum dan maksimum

Uji coba kombinasi nilai bobot inersia dilakukan untuk mengetahui kombinasi nilai bobot inersia minimum (w_{min}) dan maksimum (w_{max}) terbaik untuk menghasilkan parameter yang optimal dan solusi yang lebih baik. Pengujian ini menggunakan nilai w_{min} dan w_{max} yang berbeda antara *range* 0-1. Rancangan uji coba kombinasi bobot inersia minimum dan maksimum dapat dilihat pada Tabel 4.55.

Tabel 4.58 Pengujian kombinasi nilai bobot inersia

w_{max}	w_{min}	Nilai fitness pada percobaan ke-*i*										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	
0.6	0.2											
	0.3											
	0.4											
0.7	0.2											
	0.3											
	0.4											
0.8	0.2											
	0.3											
	0.4											

4.4.3 Pengujian terhadap nilai konstanta akselerasi

Uji coba kombinasi nilai konstanta akselerasi dilakukan untuk mengetahui kombinasi konstanta akselerasi pertama (c_1) dan konstanta akselerasi kedua (c_2) terbaik untuk menghasilkan parameter yang optimal dan solusi yang lebih baik. Pengujian ini menggunakan nilai c_1 dan c_2 yang berbeda antara *range* 1-2. Rancangan uji coba kombinasi nilai konstanta akselerasi dapat dilihat pada Tabel 4.56.

Tabel 4.59 Pengujian nilai konstanta akselerasi

c_1	c_2	Nilai fitness pada percobaan ke-*i*										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	

1	1											
	1.5											
	2											
1.5	1											
	1.5											
	2											
2	1											
	1.5											
	2											

4.4.4 Pengujian terhadap nilai konstanta pembatasan kecepatan

Uji coba terhadap nilai konstanta pembatasan kecepatan (k) untuk menghasilkan parameter yang optimal dalam menentukan *velocity clamping* dan solusi yang lebih baik. Pengujian ini menggunakan beberapa nilai k yang berbeda antara *range* 0-1. Rancangan uji coba terhadap nilai konstanta pembatasan kecepatan (k) dapat dilihat pada Tabel 4.57.

Tabel 4.60 Pengujian nilai konstanta pembatasan kecepatan

k	Nilai fitness pada percobaan ke-*i*										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
0.1											
0.2											
0.3											
0.4											
0.5											
0.6											
0.7											
0.8											
0.9											

4.4.5 Pengujian terhadap banyaknya jumlah partikel

Uji coba terhadap banyaknya jumlah partikel (*swarmsize*) untuk mengetahui jumlah partikel yang optimal dan dapat menghasilkan solusi yang lebih baik. Pengujian ini menggunakan beberapa ukuran *swarmsize* yang berbeda dengan kelipatan 5. Rancangan uji coba terhadap banyaknya jumlah partikel (*swarmsize*) dapat dilihat pada Tabel 4.58.

Tabel 4.61 Pengujian banyaknya jumlah partikel

swarmsize	**Nilai fitness pada percobaan ke-*i***										**Rata-rata fitness**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
5											
10											
20											
25											
30											

4.4.6 Pengujian terhadap jumlah maksimum iterasi

Uji coba terhadap banyaknya jumlah maksimum iterasi untuk mengetahui maksimum iterasi yang optimal dan dapat menghasilkan solusi yang lebih baik. Rancangan uji coba terhadap banyaknya jumlah maksimum iterasi dapat dilihat pada Tabel 4.59.

Tabel 4.62 Pengujian jumlah maksimum iterasi

Maks. iterasi	**Nilai fitness pada percobaan ke-*i***										**Rata-rata fitness**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
10											
20											
30											
40											
50											
60											
70											
80											

90											
100											

4.4.7 Pengujian terhadap kombinasi variasi output yang dioptimalkan

Pengujian variasi output yang dioptimalkan akan diuji dengan menggunakan salah satu kasus pengoptimalan sehingga akan mengetahui kenaikan output setelah dioptimasi. Tabel perancangan pengujian variasi output akan ditunjukkan pada Tabel 4.60

Tabel 4.63 Perancangan Pengujian Variasi Output

Kasus Optimasi	**Nilai fitness pada percobaan ke-*i***										**Rata-rata fitness**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
(Kandungan gizi yang dioptimalkan)											

BAB 5 IMPLEMENTASI

Pada bab ini membahas tentang implementasi dari perancangan *system* pemodelan komposisi pakan kambing Peranakan Etawa (PE) untuk optimasi kandungan gizi susu menggunakan *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO). Pembahasan implementasi meliputi beberapa bagian antara lain lingkungan implementasi sistem, batasan sistem, implementasi algoritma dan implementasi *user interface*.

5.1 Lingkungan Implementasi

Lingkungan implementasi berdasarkan hasil perancangan yang telah dijelaskan pada bab metodologi. Pada bagian ini juga dijelaskan mengenai batasan-batasan *system* yang digunakan. Lingkungan implementasi terdiri dari lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi memiliki spesifikasi sebagai berikut :

1. Prosesor Pentium(R) Dual Core CPU 2.3GHz
2. RAM 4.00 GB
3. *Harddisk* 500 GB
4. Monitor 14”

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi sistem ini adalah sebagai berikut :

1. Sistem operasi Windows 8.1 64 bit
2. *Microsoft Office* 2013
3. Netbeans IDE 8.0.2
4. Bahasa pemorgraman Java

5.1.3 Batasan Sistem

Terdapat beberapa batasan dalam pembuatan sistem ini, batasan-batasan tersebut adalah sebagi berikut :

1. Pakan kambing yang disediakan adalah 10 jenis bahan pakan hijauan yaitu rumput taiwan, rumput odot, rumput setaria, rumput raja, rumput B.Decumbens, kaliandra, gamal, lamtoro, turi dan tarum.

2. *Input User* berupa dua pilihan jenis pakan, kandungan gizi susu yang ingin dioptimalkan atau direndahkan serta parameter-parameter yang digunakan dalam optimasi.
3. Kandungan gizi susu berupa protein, lemak, laktosa dan berat jenis.
4. *Output* berupa kandungan nutrisi pakan serta kandungan gizi susu yang dihasilkan.
5. Metode yang digunakan adalah *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO).

5.2 Implementasi Algoritma

Implementasi algoritma dibagi menjadi 2 proses utama, yaitu implementasi algoritma *Artificial Neural Network* (ANN) dan *Particle Swarm Optimization* (PSO).

5.2.1 Implementasi algoritma *Artificial Neural Network* (ANN)

Implementasi algoritma metode ANN dibagi menjadi dua bagian utama, yaitu Proses Pelatihan ANN *Backpropagation* dan Proses Pengujian ANN *Backpropagation*.

5.2.1.1 *Proses Pelatihan ANN Backpropagation*

Proses Normalisasi

Normalisasi data merupakan proses transformasi data dimana nilai atribut data diskalakan menjadi lebih kecil dengan rentang antara 0 sampai dengan 1. Pada penelitian ini, proses normalisasi data menggunakan metode *Min-Max Normalization*. Normalisasi dilakukan dengan mencari nilai batas maksimal pada fungsi `Cari_Nilai_Max` dan minimal pada fungsi `Cari_Nilai_Min` dari data yang digunakan sesuai dengan Gambar 5.1.

```
1   public static double Cari_Nilai_Max(double[] nums) {
2           double max = nums[0];
3           for (int i = 1; i < nums.length; i++) {
4               if (nums[i] > max) {
5                   max = nums[i];
6               }
7           }
8           return max;
9       }
10  public static double Cari_Nilai_Min(double[] nums) {
11          double min = nums[0];
12          for (int i = 1; i < nums.length; i++) {
13              if (nums[i] < min) {
14                  min = nums[i];
15              }
16          }
```

```
17          return min;
18      }
19  public static double[][] Normalisasi_Data
20  (ArrayList<String> input, double[][] data, double[] min,
21  double[] max) {
22          double[][]      hasil_normalisasi      =      new
23  double[input.size()][8];
24          for (int i = 0; i < input.size(); i++) {
25              for (int j = 0; j < 8; j++) {
26                  if (data[i][j] <= 1) {
27                      hasil_normalisasi[i][j] = data[i][j];
28                  } else {
29                          hasil_normalisasi[i][j]            =
30  0.8*((data[i][j]  -  min[j])  /  (max[j]  -  min[j]))+0.1;
31  }
32              }
33          }
34          return hasil_normalisasi;
35      }
```

Gambar 5.1 *Source code* Proses Normalisasi

Penjelasan Gambar 5.1 adalah sebagai berikut :

1. Baris 3-5 merupakan pencarian nilai max pada data yang disimpan dalam array nums.
2. Baris 12-14 merupakan pencarian nilai min pada data array nums.
3. Baris 26-27 merupakan hasil dari kondisi jika data yang akan dinormalisasi memiliki nilai lebih dari 1.
4. Baris 29-30 merupakan cara menormalisasi data yang memiliki nilai lebih dari 1.
5. Baris 34 merupakan pengembalian hasil normalisasi

Proses Feedforward

Proses *feedforward* merupakan proses awal dari metode ANN, proses ini diawali dengan menentukan bobot *input* V_{ij} dan diakhiri dengan proses pencarian nilai fungsi sigmoid Y_{in}. Implementasi algoritma pada proses *feedforward* akan dijelaskan pada Gambar 5.2

```
1   public  static  double[][]  Bobot_Vij(int  iterasi,  int
2   batas_min, int batas_max,
3   int jum_hidden_layer, int data, double[][] bobot_sebelum,
4   double[][] delta_bobot) {
5           int input = 4;
6           double          bobot_Vij[][]          =          new
7   double[input][jum_hidden_layer];
8           if (iterasi == 0) {
9               if (data == 0) {
10                  for (int j = 0; j < jum_hidden_layer; j++)
11  {
```

```
12                      for (int i = 0; i < input; i++) {
13                          bobot_Vij[i][j] = (Math.random() *
14  (batas_max - batas_min)) + batas_min;
15                      }
16                  }
17              } else {
18                  for (int j = 0; j < jum_hidden_layer; j++)
19  {
20                      double hasil = 0;
21                      for (int i = 0; i < input; i++) {
22                          hasil   =   (bobot_sebelum[i][j]   +
23  delta_bobot[i][j]);
24                          bobot_Vij[i][j] = hasil;
25                      }
26                  }
27              }
28          } else {
29              if (data == 0) {
30                  for (int j = 0; j < jum_hidden_layer; j++)
31  {
32                      for (int i = 0; i < input; i++) {
33                          bobot_Vij[i][j]                    =
34  bobot_sebelum[i][j];
35                      }
36                  }
37              } else {
38                  for (int j = 0; j < jum_hidden_layer; j++)
39  {
40                      double hasil = 0;
41                      for (int i = 0; i < input; i++) {
42                          hasil   =   (bobot_sebelum[i][j]   +
43  delta_bobot[i][j]);
44                          bobot_Vij[i][j] = hasil;
45                      }
46                  }
47              }
48          }
49
50          return bobot_Vij;
51      }
52
53      public  static  double[][]  Bobot_Wjk(int  iterasi,  int
54  batas_min, int batas_max,
55              int  jum_hidden_layer,  int  data,  double[][]
56  bobot_sebelum, double[][] delta_bobot) {
57          int output = 4;
58          double          bobot_Wjk[][]          =          new
59  double[jum_hidden_layer][output];
60          if (iterasi == 0) {
61              if (data == 0) {
62                  for (int j = 0; j < jum_hidden_layer; j++)
63  {
64                      for (int k = 0; k < output; k++) {
65
```

```
                        bobot_Wjk[j][k] = (Math.random() *
(batas_max - batas_min)) + batas_min;
                    }
                }
            } else {
                for (int j = 0; j < jum_hidden_layer; j++)
{
                    double hasil = 0;
                    for (int k = 0; k < output; k++) {
                        hasil   =   (bobot_sebelum[j][k]   +
delta_bobot[j][k]);
                        bobot_Wjk[j][k] = hasil;
                    }
                }
            }
        } else {
            if (data == 0) {
                for (int j = 0; j < jum_hidden_layer; j++)
{
                    for (int k = 0; k < output; k++) {
                        bobot_Wjk[j][k]                    =
bobot_sebelum[j][k];
                    }
                }
            } else {
                for (int j = 0; j < jum_hidden_layer; j++)
{
                    double hasil = 0;
                    for (int k = 0; k < output; k++) {
                        hasil   =   (bobot_sebelum[j][k]   +
delta_bobot[j][k]);
                        bobot_Wjk[j][k] = hasil;
                    }
                }
            }
        }

        return bobot_Wjk;
    }

    public   static   double[]   Z_in(double[][]   Input_X,
double[][] Bobot_Vij, int jum_hidden_layer, int data) {

        int input = 4;

         double[] Z_in = new double[jum_hidden_layer];

        for (int j = 0; j < jum_hidden_layer; j++) {
            double hasil = 0;
            for (int i = 0; i < input; i++) {
                hasil = (Input_X[data][i] * Bobot_Vij[i][j])
+ hasil;
                Z_in[j] = hasil;
            }
```

```
120         }
121         return Z_in;
122     }
123 
124     public static double[] Y_in(double[] Z, double[][]
125 Bobot_Wjk, int jum_hidden_layer) {
126 
127         int output = 4;
128 
129         double[] Y_in = new double[output];
130 
131          for (int k = 0; k < output; k++) {
132             double hasil = 0;
133             for (int j = 0; j < jum_hidden_layer; j++) {
134                 hasil = (Z[j] * Bobot_Wjk[j][k]) + hasil;
135                 Y_in[k] = hasil;
136             }
137         }
138         return Y_in;
139     }
140 
141     public static double[] Fungsi_Sigmoid_Z(double[] Z_in,
142 int jum_hidden_layer) {
143         double[] fungsi_sigmoid = new
144 double[jum_hidden_layer];
145 
146         for (int i = 0; i < jum_hidden_layer; i++) {
147             fungsi_sigmoid[i] = 1 / (1 + Math.pow(2.71828,
148 -(Z_in[i])));
149         }
150         return fungsi_sigmoid;
151     }
152 
153     public static double[] Fungsi_Sigmoid_Y(double[] Y_in,
154 int jum_hidden_layer) {
155         int output = 4;
156         double[] fungsi_sigmoid = new double[output];
157         for (int i = 0; i < output; i++) {
158             fungsi_sigmoid[i] = 1 / (1 + Math.pow(2.71828,
159 -(Y_in[i])));
160         }
161         return fungsi_sigmoid;
162     }
```

Gambar 5.2 *Source code* Proses *Feedforward*

Penjelasan Gambar 5.2 adalah sebagai berikut:

1. Baris 12-15 merupakan proses mendapatkan nilai bobot Input V_{ij} pada baris pertama yang dibangkitkan secara *random*.
2. Baris 24-28 merupakan proses mendapatkan nilai bobot V_{ij} pada baris kedua dan seterusnya sampai kurang dari jumlah input yang diberikan.

3. Baris 12-15 merupakan proses mendapatkan nilai bobot Input Vij pada baris pertama yang dibangkitkan secara *random*.
4. Baris 21-24 merupakan proses mendapatkan nilai bobot Vij pada baris kedua dan seterusnya sampai kurang dari jumlah input yang diberikan.
5. Baris 31-34 merupakan cara mendapatkan nilai bobot Vij pada iterasi selanjutnya pada baris pertama, yaitu nilainya sama dengan nilai bobot terakhir pada iterasi sebelumnya.
6. Baris 42-43 merupakan proses mendapatkan nilai bobot Vij pada ietrasi selanjutnya pada baris kedua dan seterusnya.
7. Baris 66-67 merupakan proses membangkitkan nilai bobot Wjk pada baris pertama.
8. Baris 74-76 merupakan proses memperoleh nilai Wjk pada baris kedua dan selanjutnya.
9. Baris 85-87 merupakan proses memperoleh nilai bobot Wjk baris pertama pada iterasi selanjutnya, yaitu nilai sama dengan nilai bobot Wjk baris terakhir pada iterasi sebelumnya.
10. Baris 94-98 merupakan proses mendapatkan nilai bobot Wjk baris kedua dan seterusnya pada iterasi selanjutnya.
11. Baris 115-118 merupakan proses untuk mendapatkan nilai Z_in, yaitu perkalian antara data input dengan bobot Vij.
12. Baris 133-135 merupakan proses mendapatkan nilai Yin yaitu perkalian antara nilai Z dengan bobot_Wjk.
13. Baris 147-149 merupakan proses mendapatkan nilai Z yang membutuhkan nilai Z_in dan nilai konstanta e yaitu 2.71828.
14. Baris 158-159 merupakan proses mendapatkan nilai Y yaitu dengan membutuhkan nilai data Y_in data konstanta e yang nilainy sudah ditetapkan pakar.

Proses Backpropagation

Proses *Backpropagation* merupakan proses lanjutan dari proses *feedforward*, proses ini dimulai dari mencari nilai *output* sampai dengan mendapatkan nilai MSE *error*. Implementasi proses *backpropagation* akan dijelaskan pada Gambar 5.3.

```
1  public static double[] Error_Output(double[][] target,
2  double[] Y, int data) {
3
4          int output = 4;
5
6          double error_output[] = new double[output];
```

```
7
8           for (int i = 0; i < output; i++) {
9               error_output[i] = (target[data][i] - Y[i]) *
10  Y[i] * (1 - Y[i]);
11          }
12
13          return error_output;
14      }
15
16      public static double[][] Delta_Wjk(double[]
17  error_output, double[] Z, int jum_hidden_layer, double
18  learning_rate) {
19
20          int output = 4;
21
22          double delta_Wjk[][] = new
23  double[jum_hidden_layer][output];
24
25          for (int j = 0; j < jum_hidden_layer; j++) {
26              for (int k = 0; k < output; k++) {
27                  delta_Wjk[j][k] = (2*(learning_rate *
28  error_output[k] * Z[j]));
29  //                  System.out.println(learning_rate + " * "
30  + error_output[k] + " * " + Z[j]);
31              }
32          }
33
34          return delta_Wjk;
35      }
36
37      public static double[] Jumlah_Delta_Input(double[]
38  error_output, double[][] bobot_Wjk, int jum_hidden_layer) {
39
40          int output = 4;
41
42          double jum_delta_input[] = new
43  double[jum_hidden_layer];
44
45          for (int i = 0; i < jum_hidden_layer; i++) {
46              double hasil = 0;
47              for (int j = 0; j < output; j++) {
48                  hasil = (error_output[j] * bobot_Wjk[i][j])
49  + hasil;
50  //                  System.out.println(error_output[j] + " *
51  " + bobot_Wjk[i][j] + " + " + hasil);
52              }
53              jum_delta_input[i] = hasil;
54  //              System.out.println("= " + jum_delta_input[i]);
55          }
56
57          return jum_delta_input;
58      }
59      public static double[] Informasi_Error(double[]
60  jum_delta_input, double[] Z, int jum_hidden_layer) {
```

```
61
62          double informasi_error[] = new
63  double[jum_hidden_layer];
64
65          for (int i = 0; i < jum_hidden_layer; i++) {
66              informasi_error[i] = jum_delta_input[i] * Z[i]
67  * (1 - Z[i]);
68          }
69
70          return informasi_error;
71      }
72
73      public static double[][] Delta_Vij(double[]
74  informasi_error, double[][] X,
75              int jum_hidden_layer, double learning_rate, int
76  data) {
77
78          int input = 4;
79
80          double[][] delta_Vij = new
81  double[input][jum_hidden_layer];
82
83          for (int j = 0; j < jum_hidden_layer; j++) {
84              for (int i = 0; i < input; i++) {
85                  delta_Vij[i][j] = 2*(learning_rate *
86  informasi_error[j] * X[data][i]);
87              }
88          }
89
90          return delta_Vij;
91      }
92
93      public static double Hitung_Error(double[][] target,
94  double[] Y, int data) {
95
96          double error = 0;
97          int output = 4;
98          double jum_output = 4;
99
100         double hasil = 0;
101         for (int j = 0; j < output; j++) {
102             hasil = (target[data][j] - Y[j]) + hasil;
103         }
104         error = (1 / jum_output) * hasil;
105
106         return error;
107     }
108
109     public static double Hitung_MSE(double[] error) {
110
111         double MSE = 0;
112         for (int i = 0; i < error.length; i++) {
113             MSE += (Math.pow(error[i], 2) / error.length);
114         }
```

```
115             return MSE;
116         }
```

Gambar 5.3 Algoritma *Backpropagation*

Penjelasan Gambar 5.3 adalah sebagai berikut :

1. Baris 9-10 merupakan cara yang dilakukan untuk memperoleh data error output, yaitu dengan perkalian selisih antara target output yang sudah ditentukan di awal, dengan nilai pada fungsi sigmoid Y.
2. Baris 27-30 merupakan cara untuk mendapatkan nilai delta_W_{jk} yaitu membutuhkan data learning rate, nilai error output serta nilai sinyal input Z.
3. Baris 53-53 merupakan proses untuk mendapatkan nilai jumlah delta input yang panjangnya sama dengan jumlah hidden layer.
4. Baris 66-67 merupakan proses untuk mendapatkan nilai informasi error yang didapatkan dari perkalian antara Z dengan jumlah data input.
5. Baris 85-86 merupakan proses untuk mendapatkan nilai delta_V_{ij}, yaitu terdapat perulangan sebanyak jumlah hidden layer dan sebanyak input.
6. Baris 104-105 merupakan proses untuk mendapatkan nilai error yaitu denganmembutuhkan data jum_output dan hasil. Hasil didapatkan dari total selisih antara nilai target dengan nilai Y.
7. Baris 113-114 merupakan proses untuk mendapatkan nilai MSE yang didaptkan dari total jumlah kuadrat nilai error dibagi dengan panjang error.

5.2.1.2 *Proses Pengujian ANN Backpropagation*

Proses pengujian ANN *Backpropagation* hanya melakukan proses feedforward yang ditambah dengan proses denormalisasi pada akhir proses feedforward, proses denormalisasi merupakan proses pengembalian nilai output yang pada proses awal sudah ternormalisasi. Implementasi proses denormalisasi akan ditunjukkan pada Gambar 5.4.

Proses Denormalisasi

```
1   public static double[] Denormalisasi(double[][] input_x,
2   double[] Y, int data) {
3
4           int length = input_x.length;
5           double[] denormalisasi = new double[4];
6
7           Kandungan_Gizi_Susu KGS = new Kandungan_Gizi_Susu();
8
9   KGS.setMinMaxProtein(Cari_Nilai_Min(KGS.proteinArray),
10  Cari_Nilai_Max(KGS.proteinArray));
11          KGS.setMinMaxLemak(Cari_Nilai_Min(KGS.lemakArray),
12  Cari_Nilai_Max(KGS.lemakArray));
13
```

```
14
15  KGS.setMinMaxLaktosa(Cari_Nilai_Min(KGS.laktosaArray),
16  Cari_Nilai_Max(KGS.laktosaArray));
17
18  KGS.setMinMaxDensity(Cari_Nilai_Min(KGS.densityArray),
19  Cari_Nilai_Max(KGS.densityArray));
20
21          double minProtein = KGS.getMinProtein();
22          double minLemak = KGS.getMinLemak();
23          double minLaktosa = KGS.getMinLaktosa();
24          double minDensity = KGS.getMinDensity();
25          double[]    minData    =    {minProtein,    minLemak,
26  minLaktosa, minDensity};
27
28          double maxProtein = KGS.getMaxProtein();
29          double maxLemak = KGS.getMaxLemak();
30          double maxLaktosa = KGS.getMaxLaktosa();
31          double maxDensity = KGS.getMaxDensity();
32          double[]    maxData    =    {maxProtein,    maxLemak,
33  maxLaktosa, maxDensity};
34
35              for (int j = 0; j < 4; j++) {
36  denormalisasi[j]=          (((maxData[j]-minData[j])*(Y[j]-
37  0.1))/0.8)+minData[j
38              }
39
40          return denormalisasi;
41      }
```

Gambar 5.4 Proses denormalisasi

Penjelasan Gambar 5.4 akan dijelaskan sebagai berikut:

1. Baris 21-25 merupakan proses pencarian nilai minimal dari setiap data kandungan gizi susu.
2. Baris 28-33 merupakan proses pencarian nilai maksimal dari setiap kandungan gizi susu.
3. Baris 35-37 merupakan proses untuk mendapatkan nilai denormalisasi yaitu dengan membutuhkan data Y, nilai maksimal dari data, dan nilai minimal dari data.

5.2.2 Implementasi algoritma *Particle Swarm Optimization* (PSO)

Proses Optimasi digunakan algoritma PSO, algoritma ini akan melakukan optimasi sesuai dengan output yang diinginkan oleh user. Pada proses ini , kumpulan GBEST adalah solusi terbaik dalam pemberian pakan untuk mendapatkan output yang diinginkan. Berikut adalah langkah-langkah implementasi pada proses optimasi PSO :

Proses Inisialisasi Partikel

Proses pertama yang dilakukan dalam algoritma PSO ini merupakan pembangkitan partikel pada iterasi ke 0 seperti yang ditunjukkan pada Gambar 5.5.

```
1   public static double[][] Inisialisasi_Partikel(double[]
2   batas_bawah, double[] batas_atas,int swarm_size, int
3   dimensi) {
4         partikelArgument = new double[swarm_size][dimensi +
5   1];//+1 untuk tambahan fitnes
6         Partikel p = new Partikel ();
7
8         for (int i = 0; i < swarm_size; i++) {
9             double[] X = new double[dimensi];
10            double[] V0 = new double[dimensi];
11            for (int j = 0; j < dimensi; j++) {
12
13                partikelArgument[i][j] = (BatasBawahRand[j]
14  + (Math.random() * (BatasAtasRand[j] - BatasBawahRand[j])));
15                Position location = new Position(X);
16                p.setPosition(location);
17
18                V0[j] = 0;
19                Velocity velocity = new Velocity(V0);
20                p.setVelocity(velocity);
21
22            }
23        }
24
25        return partikelArgument;
26    }
```

Gambar 5.5 Proses Inisialisasi Partikel

Penjelasan dari Gambar 5.5 adalah sebagai berikut:

1. Baris 11-14 merupakan proses untuk membangkitkan partikel secara random berdasarkan batas bawah dan batas atas.
2. Baris 15 merupakan proses setting posisi berdasarkan partikel yang sudah dibangkitkan.
3. Baris 18-20 merupakan proses setting velocity untuk iterasi ke-0.

Proses Pencarian nilai Fitness

Fitness didapatkan setelah proses ANN yang *feedforward* dilakukan sampai menemukan nilai denormalisasi masing-masing *output* sehingga dapat dicari nilai *fitness* sesuai dengan persamaan 2.18. Proses pencarian nilai *fitness* dijelaskan pada Gambar 5.6.

```
1   public static double[] Hitung_Fitness(int optimal, int
2   minimal, int swarm_size, double[] denormalisasi) {
3
```

```
4           double[] fitness = new double[swarm_size];
5           for (int j = 0; j < swarm_size; j++) {
6               fitness[j] = denormalisasi[optimal-1] + (1 /
7   denormalisasi[minimal-1]);
8               partikelArgument[j][4] = fitness[j];
9           }
10          return fitness;
11      }
12
```

Gambar 5.6 Proses Hitung Fitness

1. Baris 6-8 merupakan proses pencarian fitness berdasarkan nilai output yang dioptimalkan dan diminimalkan. Fitness dapat dihitung jika sudah melalui proses denormalisasi.

Proses Memperbarui pBest

P-Best merupakan *local best* yang didapatkan setelah mendapatkan nilai *fitness* yang sudah diurutkan. Nilai fitness terbesar dijadikan rekomendasi pBest pada iterasi t. Proses memperbarui pBest ditunjukkan pada Gambar 5.7.

```
1   if (iterasi > 0) {
2               for (int i = 0; i < swarm_size; i++) {
3                   if (Hitung_fitness[i] > pbest[i]) {
4                       pbest[i] = Hitung_fitness[i];
5           } else {
6               for (int i = 0; i < swarm_size; i++) {
7                   pbest[i] = Hitung_fitness[i];
8               }
9           }
10
11          return pbest;
12      }
```

Gambar 5.7 Proses Pencarian PBest

Penjelasan Gambar 5.7 adalah sebagai berikut:

1. Baris 1-4 merupakan proses pencarian nilai pbest dengan iterasi>0, jika nilai fitness lebih dari pbest sebelumnya, maka nilai pbest saat ini merupakan nilai fitness tersebut.
2. Baris 6-8 merupakan pencarian nilai pbest jika iterasi = 0.

Proses Mencari gBest

G-Best merupakan *global best* yang didapatkan setelah melakukan proses pencarian pBest. G-best didapatkan dari pBest yang memiliki nilai terbesar. Proses pencarian G-best ditunjukkan pada Gambar 5.8.

```
1   public static double[] update_GBest(double[][] partikel,
2   double[] update_PBest, int dimensi) {
3           double nilaiSort = bubblesort(update_PBest);
```

```
4           Nilaiupdate_GBest = update_PBest;
5
6           for (int i = 0; i < nilaiSort; i++) {
7               if (nilaiSort == partikel[i][4]) {
8   nilaiAkhir[countIndex]=partikel[i];
9                   countIndex++;
10                  for (int j = 0; j < dimensi + 1; j++) {
11                      Nilaiupdate_GBest[i] = partikel[i][j];
12                  }
13
14              }
15
16          }
17          return Nilaiupdate_GBest;
18      }
```

Gambar 5.8 Proses Pencarian G_Best

Penjelasan Gambar 5.8 adalah sebagai berikut:

1. Baris 3 merupakan proses sorting dari nilai fitness pada pbest.
2. Baris 6-14 merupakan proses jika `nilaiSort` yaitu fitness tertinggi sudah didapatkan, maka akan diambil nilai *index*-nya yang kemudian akan dijadikan gbest.

Proses Update Kecepatan

Setiap partikel memiliki kecepatan dalam mendapatkan masing-masing fitness. Kecepatan dijadikan sebagai acuan untuk mendapatkan posisi partikel pada iterasi t. Proses mencari kecepatan ditunjukkan pada Gambar 5.9.

```
1       public static double[][] Kecepatan(double[][] partikel,
2   double c1, double c2, int iterasi, double[] inersia, int
3   swarm_size, int dimensi, double[] Vmax, double[]
4   update_PBest, double[] update_GBest, double[][]
5   Cetak_Posisi_pbest) {
6           double r1 = Math.random();
7           double r2 = Math.random();
8           System.out.println("r1 : " + df.format(r1));
9           System.out.println("r2 : " + df.format(r2));
10          newVel = new double[swarm_size][dimensi];
11
12          for (int i = 0; i < swarm_size; i++) {
13
14              for (int j = 0; j < dimensi; j++) {
15                  if (iterasi == 1) {
16                      newVel[i][j] = (inersia[iterasi] *
17  p.getVelocity().getPos()[i][j]) + (r1 * c1) *
18  (pos_Pbest[i][j] - partikel[i][j]) + (r2 * c2) *
19  (Nilaiupdate_GBest[Nilaiupdate_GBest.length - 1] -
20  partikel[i][j]);
21                  } else if (iterasi > 1) {
22
23
```

```
24                      newVel[i][j]   =   (inersia[iterasi]   *
25  p.getVelocity().getPos()[i][j])    +    (r1    *    c1)    *
26  (update_PBest[j]   -   partikel[i][j])   +   (r2   *   c2)   *
27  (Nilaiupdate_GBest[Nilaiupdate_GBest.length    -    1]    -
28  partikel[i][j]);
29
30                      //pembatasan kecepatan
31                      if (newVel[i][j] > Vmax[j]) {
32                          newVel[i][j] = Vmax[j];
33                      } else {
34                          newVel[i][j] = newVel[i][j];
35                      }
36                  }
37              }
38
39          }
40          return newVel;
        }
```

Gambar 5.9 Proses Update Kecepatan

Penjelasan Gambar 5.9 adalah sebagai berikut:

1. Baris 15-20 proses mencari nilai kecepatan jika iterasi = 1, sehingga pbest iterasi sebelumnya merupakan partikel itu sendiri.
2. Baris 22-27 proses mencari nilai kecepatan jika iterasi lebih dari 1.
3. Baris 29-34 proses pembatasan nilai kecepatan berdasarkan Vmax.

Proses Update Posisi

Posisi partikel ditentukan berdasarkan penjumlahan antara nilai kecepatan dan batas vmax jika kecepatan melebihi vmax yang sudah ditentukan, agar posisi partikel tidak melampaui dari batas atas dan batas bawah yang sudah ditentukan. Proses *update* posisi ditunjukkan pada Gambar 5.10.

```
1   public   static   double[][]   Posisi(double[][]   newVel,
2   double[][] partikel, int dimensi, int swarm_size) {
3           //memperbarui posisi partikel
4           double[][] newPos = new double[swarm_size][dimensi
5   + 1];//ditambah fitness
6           for (int i = 0; i < swarm_size; i++) {
7               for (int j = 0; j < dimensi; j++) {
8                   newPos[i][j]     =     partikel[i][j]     +
9   newVel[i][j];
10                  //pembatasan ruang pencarian atau dimensi
11  partikel
12                  if (newPos[i][j] < BatasBawahRand[j]) {
13                      newPos[i][j] = BatasBawahRand[j];
14                  } else if (newPos[i][j] > BatasAtasRand[j])
15  {
16                      newPos[i][j] = BatasAtasRand[j];
17                  } else {
18                      newPos[i][j] = newPos[i][j];
```

```
19                      }
20                  }
21
22              }
23              Position loc = new Position(newPos);
24              p.setPosition(loc);
25              return newPos;
26          }
```

Gambar 5.10 Proses Update Posisi

Penjelasan Gambar 5.10 adalah sebagai berikut:

1. Baris 7-9 merupakan proses mendapatkan nilai posisi baru yang didapatkan dari penjumlahan antara nilai partikel dengan nilai kecepatan
2. Baris 12-18 merupakan proses pembatasan nilai posisi yang dibatasi oleh nilai batas bawah dan nilai batas atas.

Proses Random Injection

Random injection dilakukan untuk mengatasi konvergensi dini. Partikel akan diinjeksi dengan partikel baru sebanyak 30% partikel dari ukuran *swarmsize*. Proses *random injection* ditunjukkan pada Gambar 5.11.

```
1   public double [][]random_Injection(double [] Hitung_Fitness,
2   double[]    update_PBest,double[]    batas_bawah,    double[]
3   batas_atas,  int swarm_size, int dimensi, int iterasi){
4   int jumlah_injeksi = 30/100*swarm_size;
5            newPosInjec = new double[swarm_size][dimensi];
6                newVelInjec = new double[swarm_size][dimensi];
7            for (int i = 0; i < dimensi; i++) {
8               BatasBawahRand[i] = batas_bawah[i];
9               BatasAtasRand[i] = batas_atas[i];
10           }
11           for (int i = 0; i < swarm_size; i++) {
12               bubblesort(Hitung_Fitness);
13  for (int j = 0; j < jumlah_injeksi; j++){
14        if(update_PBest[iterasi-1][i]==Hitung_Fitness[i]){
15                                for (int k = 0; k < dimensi;
16  k++) {
17                                    partikelArgument[i][k]    =
18  BatasBawahRand[k]  +  Math.random()  *  (BatasAtasRand[k]  -
19  BatasBawahRand[k]) ;
20
21                                }
22                                }
23                            }
            }return partikelArgument;
        }
```

Gambar 5.11 Proses *Random Injection*

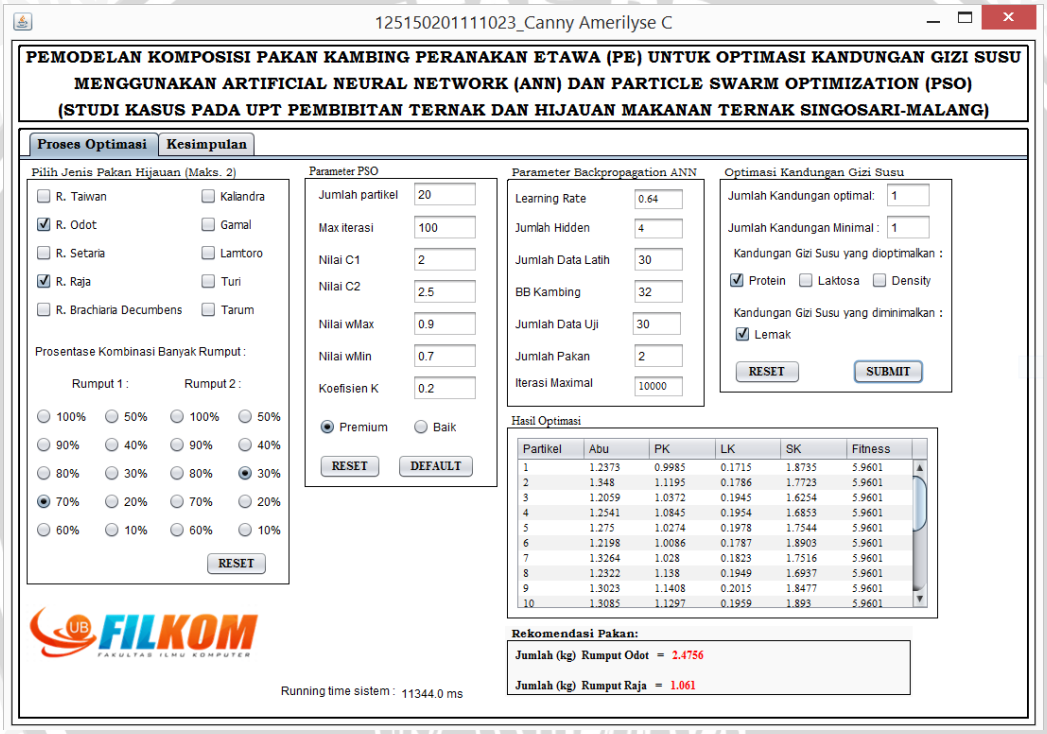
Penjelasan Gambar 5.11 adalah sebagai berikut:

1. Baris 11 untuk sorting nilai fitness.

2. Baris 12 adalah kondisi dimana nilai pbest iterasi sebelum sama dengan nilai fitness iterasi saat ini.
3. Baris 13-18 untuk menginisialisasi partikel secara *random*

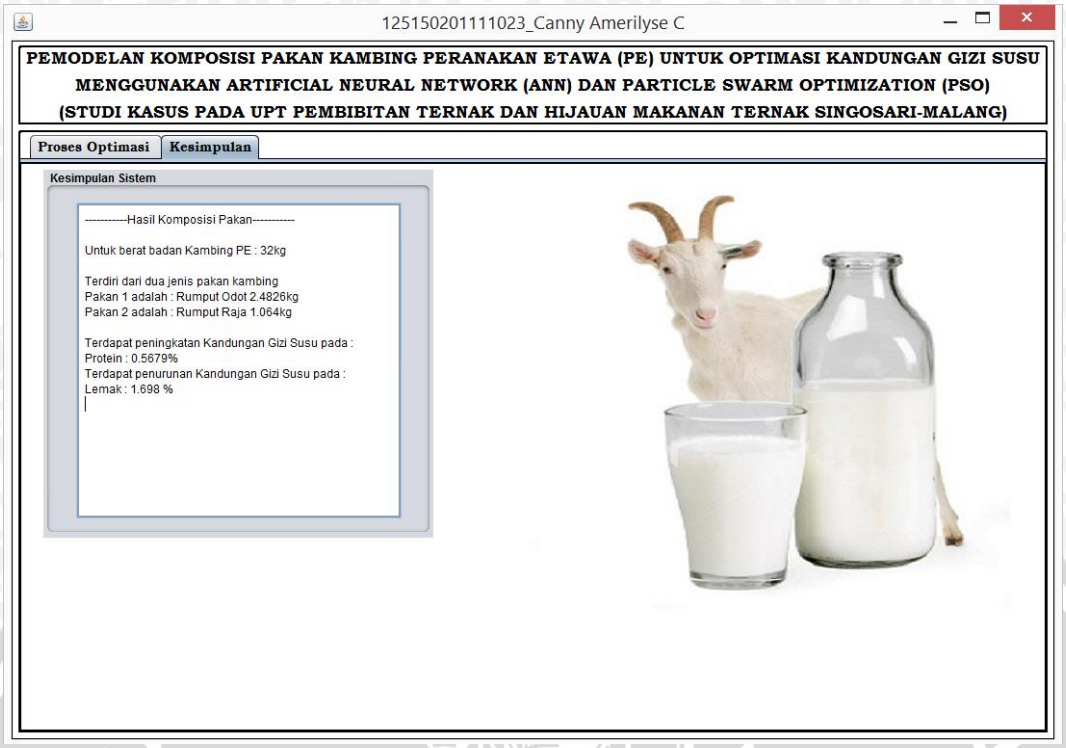
5.3 Implementasi *User Interface*

Antarmuka sistem terdiri dari halaman data sistem, dimana user dapat memberikan input pilihan pakan, parameter ANN dan parameter PSO yang akan diberikan. Pada halaman ini, user akan mendapatkan hasil terakhir dari system yaitu fitness terbaik dari optimasi serta jumlah pakan yang direkomendasikan oleh system dalam ukuran kilogram (kg). Tampilan antarmuka program ditunjukkan dalam Gambar 5.11 berikut:



Gambar 5.12 Implementasi Halaman Optimasi

Pada halaman berikutnya merupakan halaman hasil kesimpulan dari sistem. Halaman kesimpulan didapatkan dari hasil output pada optimasi. Pada halaman hasil kesimpulan akan diberikan beberapa informasi terkait dengan komposisi pakan yang akan direkomendasikan dalam mengoptimasi kandungan gizi susu kambing sesuai dengan berat badan kambing yang dimasukkan oleh user sebagai data masukkan. Tampilan halaman hasil kesimpulan sistem ditunjukkan dalam Gambar 5.12 berikut:



Gambar 5.13 Implementasi halaman hasil Kesimpulan

Halaman hasil kesimpulan ini akan memberikan keterangan pada *user* mengenai hasil komposisi pakan untuk berat badan kambing sesuai data masukkan dari *user*. Hasil komposisi pakan akan terdiri dari nama pakan serta jumlah pakan yang direkomendasikan dalam bentuk kilogram (kg). Hasil kenaikan nilai kandungan gizi susu juga akan diinformasikan sesuai dengan data masukkan kandungan gizi optimal yang dimasukkan *user*.

BAB 6 PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai pengujian serta analisis dari *system* yang sudah diimplementasi. Proses pengujian akan dilakukan berdasarkan skenario pengujian yang sudah dijelaskan pada bab perancangan.

6.1 Pengujian

Pengujian yang dilakukan berdasarkan skenario adalah sebagai berikut :

1. Pengujian terhadap maksimal iterasi pada ANN
2. Pengujian terhadap kombinasi nilai bobot inersia minimal dan maksimal
3. Pengujian terhadap nilai konstanta akselerasi
4. Pengujian terhadap nilai konstanta pembatasan kecepatan
5. Pengujian terhadap banyaknya jumlah partikel
6. Pengujian terhadap jumlah maksimum iterasi
7. Pengujian terhadap variasi output yang digunakan

6.2 Hasil Pengujian

Berdasarkan skenario pengujian yang sudah dirancang, hasil pengujian akan terdapat perbandingan pada masing-masing grafik, yaitu perbandingan dari kualitas susu Premium serta kualitas susu Baik, dari perbandingan tersebut akan terlihat parameter-parameter yang menghasilkan *fitness* optimal. Pada sub-bab hasil pengujian akan ditampilkan hasil pengujian dari kualitas susu Premium, sedangkan kualitas susu Baik akan ditampilkan pada Lampiran. Maka hasil yang diperoleh dari pengujian tersebut adalah sebagai berikut:

6.2.1 Hasil Pengujian terhadap maksimal iterasi pada data latih ANN

Pengujian iterasi maksimal pada data latih ANN menggunakan data sebagai berikut :

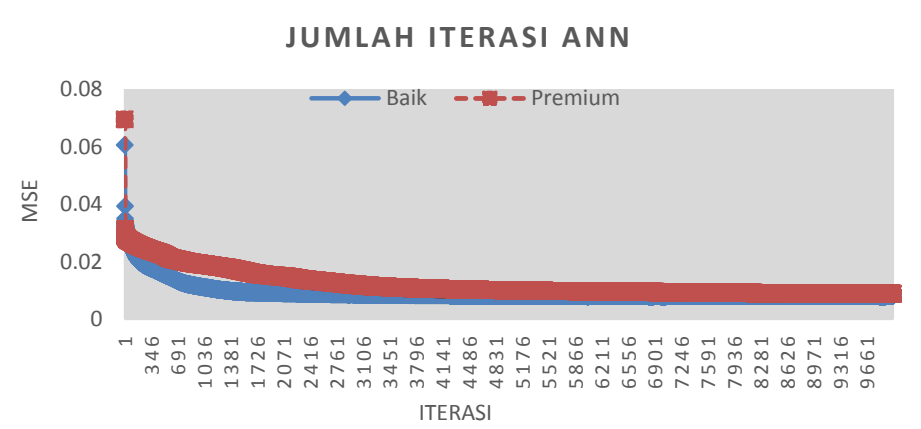
1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.
3. *Hidden layer* sejumlah 4.

Pengujian ini akan menghasilkan nilai MSE. Hasil pengujian ditunjukkan pada Tabel 6.1. Tabel hasil pengujian selengkapnya akan terdapat pada Lampiran 3.

Tabel 6.1 Hasil pengujian maksimal iterasi data latih ANN

Jumlah Iterasi	Nilai fitness pada percobaan ke-*i*					Rata-rata MSE
	1	2	3	4	5	
10	0.0271	0.0291	0.03	0.029	0.0289	0.02882
20	0.0267	0.0286	0.0295	0.0285	0.0283	0.02832
50	0.0261	0.0283	0.0289	0.0274	0.0269	0.02752
100	0.0255	0.0281	0.0286	0.0264	0.0252	0.02676
150	0.0247	0.0279	0.0282	0.0256	0.0238	0.02604
200	0.0238	0.0277	0.0279	0.0249	0.0229	0.02544

Jumlah maksimum iterasi data latih ANN mempengaruhi iterasi yang dipakai dalam optimasi, sehingga semakin kecil nilai MSE yang dihasilkan, maka akan semakin optimal hasil optimasi yang dilakukan. Grafik hasil pengujian perbandingan jumlah data uji dan data latih digambarkan pada Gambar 6.1. Dari grafik tersebut dapat disimpulkan bahwa semakin besar iterasi, akan menghasilkan nilai MSE yang semakin kecil. Dari Gambar 6.1 didapatkan nilai MSE terkecil untuk Premiun adalah 0.0088 pada iterasi 10000 sedangkan Baik adalah 0.008 pada iterasi 10000.



Gambar 6.1 Grafik pengujian iterasi maskimum data latih ANN

6.2.2 Hasil Pengujian Kombinasi Nilai Bobot Inersia

Pengujian kombinasi nilai bobot inersia yaitu w_{max} dan w_{min} yang mempengaruhi kecepatan yang ditempuh oleh setiap partikel dalam memperoleh fitness optimal. Pengujian nilai kombinasi inersia dilakukan dengan menggunakan data sebagai berikut:

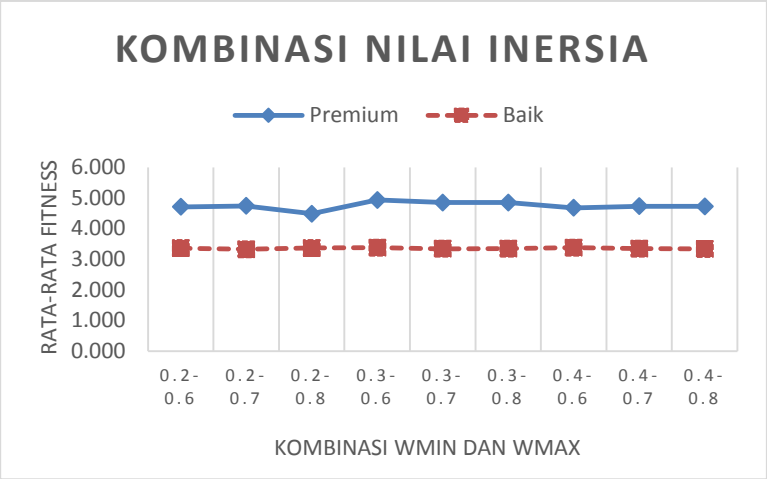
1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.

3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN = 10000 iterasi.
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium
7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai c_1 = 2 dan c_2 = 2
11. Nilai k = 0.6
12. Jumlah partikel = 10 partikel.
13. Jumlah iterasi maksimal PSO = 10 iterasi.

Hasil pengujian kombinasi inersia untuk kualitas susu Premium ditunjukkan pada Tabel 6.2. Hasil pengujian untuk kualitas susu Baik akan dilampirkan pada Lampiran 4. Grafik pengujian ditunjukkan pada Gambar 6.2.

Tabel 6.2 Hasil Pengujian kombinasi nilai bobot inersia

w_{min}	w_{max}	Nilai fitness pada percobaan ke-*i*					Rata-rata fitness
		1	2	3	4	5	
0.2	0.6	3.9277	3.8856	4.5854	4.0206	7.9499	4.7115
	0.7	6.9431	3.6191	8.0015	5.8331	4.1677	4.7409
	0.8	3.7336	4.0688	6.1683	3.9844	3.5716	4.4852
0.3	0.6	4.7312	4.3224	4.1684	7.132	6.3426	4.9347
	0.7	4.2349	4.0387	6.2308	4.2428	3.9874	4.8487
	08	4.6881	8.4688	4.2619	4.9668	5.2961	4.8503
0.4	0.6	5.3974	3.913	5.449	7.4536	4.1406	4.6761
	0.7	4.225	5.5045	4.9692	5.798	3.9899	4.7281
	0.8	3.4157	4.8665	4.553	7.6621	4.0924	4.7241



Gambar 6.2 Grafik pengujian Kombinasi Nilai inersia

Pada Gambar 6.2 menunjukkan bahwa kombinasi nilai inersia dengan menghasilkan nilai rata-rata fitness tertinggi terdapat pada kombinasi w_{min} = 0.3 dan w_{max} = 0.6. Bobot inersia menjadi pengontrol dari daya eksplorasi dan eksploitasi partikel dalam mencari solusi yang optimal. Nilai *w* yang besar akan membuat kecepatan partikel akan bertambah pesat dan menuju kecepatan maksimum. Hal ini menyebabkan eksplorasi dengan peningkatan keberagaman. Nilai *w* yang kecil mengakibatkan eksploitasi lokal. Ketika partikel cenderung eksplorasi, partikel akan bergerak menuju wilayah yang baru dan kehilangan kesempatan untuk eksploitasi wilayah-wilayah yang dikunjungi. Ketika partikel cenderung melakukan eksploitasi, partikel akan lebih fokus untuk mencari solusi pada wilayah tersebut, padahal belum dapat dipastikan bahwa wilayah tersebut akan menghasilkan solusi yang optimal dan kehilangan kesempatan untuk eksplorasi wilayah-wilayah lain yang kemungkinan menghasilkan solusi.

6.2.3 Hasil Pengujian terhadap nilai konstanta akselerasi

Pengujian nilai konstanta akselerasi atau c1 dan c2 yang mempengaruhi kecepatan yang ditempuh oleh setiap partikel dalam memperoleh fitness optimal. Pengujian nilai konstanta akselerasi dilakukan dengan menggunakan data sebagai berikut :

1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.
3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN adalah 10000
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium

7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai w_{min} = 0.3 dan w_{max} = 0.6
11. Nilai k = 0.6
12. Jumlah partikel = 10 partikel.
13. Jumlah iterasi maksimal PSO = 10 iterasi.

Hasil pengujian kombinasi inersia untuk kualitas susu Premium ditunjukkan pada Tabel 6.3. Tabel hasil pengujian nilai konstanta akselerasi secara lengkap untuk kualitas susu Premium dan Baik akan dilampirkan pada Lampiran 5. Sedangkan grafik pengujian ditunjukkan pada Gambar 6.4.

Tabel 6.3 Hasil Pengujian nilai konstanta akselerasi

c_1	c_2	Nilai fitness pada percobaan ke-*i*					Rata-rata fitness
		1	2	3	4	5	
1	1	4.3727	6.3504	3.6277	3.9439	3.8144	4.3118
	1.5	4.5981	3.831	5.9101	4.1209	4.256	4.4687
	2	4.8374	4.1436	4.096	4.0405	4.7421	4.3765
1.5	1	4.1157	4.5687	4.056	4.6167	4.0892	4.2176
	1.5	4.0358	4.0449	3.3778	4.2673	3.781	4.0616
	2	3.4355	3.8107	4.2188	4.2229	6.6468	4.3584
2	1	4.0122	4.1035	4.1605	3.5717	6.1426	4.4409
	1.5	5.3096	3.566	3.597	4.3381	6.4134	4.3274
	2	3.338	5.8867	3.9761	4.3407	4.1932	4.4434

Konstanta akselerasi mempengaruhi jarak maksimum yang dapat diambil oleh sebuah partikel dalam sebuah iterasi. c_1 mengatur jarak yang dipengaruhi oleh memori posisi *personal best* dari partikel tersebut. Sedangkan c_2 mengatur jarak maksimum yang dipengaruhi oleh partikel *global best.* Hasil pengujian konstanta kecepatan yang mendapatkan nilai rata-rata fitness tertinggi adalah c_1 = 1 dan c_2 = 1.5 dengan rata-rata fitness untuk kualitas susu Premium adalah 4.4687, sedangkan untuk kualitas susu Baik menghasilkan fitness sebesar 3.7209. Untuk sebuah sistem PSO , area pencarian partikel akan konvergen jika memenuhi kondisi $1 > w > \frac{1}{2}(c_1 + c_2) - 1 \geq 0$, serta jika nilai w, c_1 dan c_2 yang terpilih memenuhi kondisi tersebut, maka sistem dipastikan akan berada di kondisi

ekulibrium. Sebuah kondisi yang menunjukkan sistem akan menemukan solusi yang optimal (Engelbrecht A. P., 2007). Berikut ini pembuktian kondisi ekuilibrium dengan nilai c_1 dan c_2 yang terpilih:

Iterasi ke-0

Menghitung bobot inersia berdasarkan Persamaan 2.26.

$$w(0) = (0.6 - 0.3) * \left(\frac{10 - 0}{10}\right) + 0.3 = 0.6$$

$$1 > 0.6 > \frac{1}{2}(1 + 1.5) - 1 \geq 0$$

$$1 > 0.6 > 0.25 \geq 0$$

Iterasi ke-10

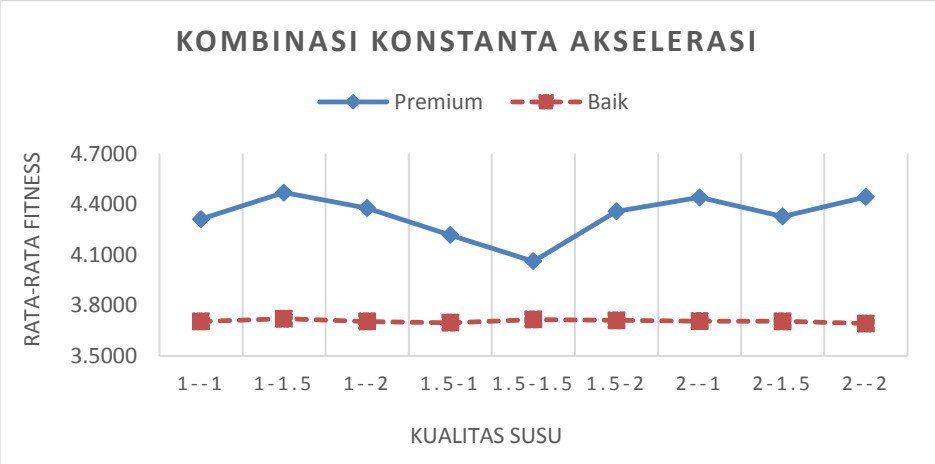
Menghitung bobot inersia berdasarkan Persamaan 2.26

$$w(0) = (0.6 - 0.3) * \left(\frac{10 - 10}{10}\right) + 0.3 = 0.3$$

$$1 > 0.3 > \frac{1}{2}(1 + 1.5) - 1 \geq 0$$

$$1 > 0.3 > 0.25 \geq 0$$

Grafik pengujian ditunjukkan oleh Gambar 6.3.



Gambar 6.3 Grafik pengujian nilai Konstanta Kecepatan

6.2.4 Hasil Pengujian terhadap nilai konstanta pembatasan kecepatan

Pengujian nilai konstanta pembatasan kecepatan yang mempengaruhi kecepatan yang ditempuh agar kecepatan partikel tidak melebihi kecepatan maksimal yang sudah ditentukan dalam memperoleh fitness optimal. Pengujian nilai konstanta pembatasan dilakukan dengan menggunakan data sebagai berikut:

1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.

3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN adalah 10000
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium
7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai w_{min} = 0.3 dan w_{max} = 0.6
11. Nilai c_1 = 1 dan c_2 =1.5
12. Jumlah partikel = 10 partikel.
13. Jumlah iterasi maksimal PSO = 10 iterasi.

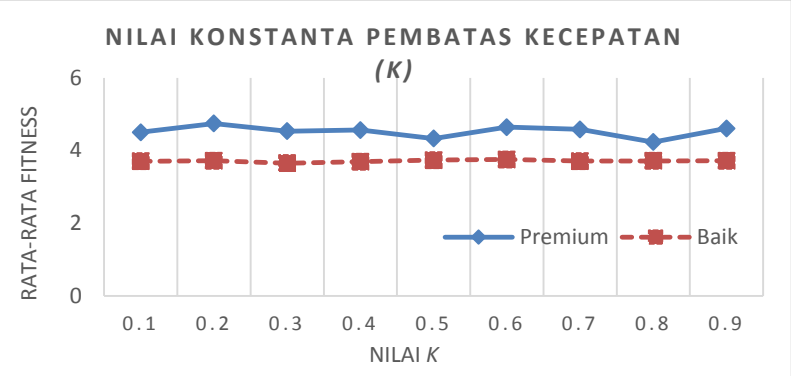
Hasil pengujian konstanta pembatasan kecepatan untuk kualitas susu Premium ditunjukkan pada Tabel 6.4. Tabel hasil pengujian nilai konstanta pembatasan kecepatan untuk kualitas susu Premium dan Baik, secara lengkap akan dilampirkan pada Lampiran 6.

Tabel 6.4 Hasil Pengujian nilai konstanta pembatasan kecepatan

k	Nilai fitness pada percobaan ke-*i*					Rata-rata fitness
	1	2	3	4	5	
0.1	4.2179	3.8941	3.5788	3.6121	6.8143	4.50256
0.2	4.0627	5.8315	4.7374	4.0132	3.6645	4.7455
0.3	4.9393	6.2623	3.3586	4.4449	4.0442	4.53183
0.4	4.8989	3.689	3.8247	3.7322	3.983	4.56577
0.5	4.0612	7.7108	3.9321	4.1066	3.645	4.3301
0.6	3.338	5.8867	3.9761	6.3407	4.1932	4.64342
0.7	4.0814	3.782	4.0023	3.8355	3.7548	4.58083
0.8	4.0287	3.9241	4.4896	6.3355	3.5075	4.23399
0.9	4.1379	6.3442	4.189	4.1587	4.0008	4.60942

Pada Gambar 6.4 menunjukkan bahwa nilai konstanta pembatasan kecepatan (*k*) yang dapat menghasilkan nilai rata-rata fitness tertinggi adalah 0.6 dengan rata-rata fitness untuk kualitas susu Premium adalah 4.6434, sedangkan kualitas susu Baik menghasilkan fitness sebesar 3.7518. Nilai konstanta tersebut akan

digunakan dalam pengujian selanjutnya. Semakin kecil nilai konstanta *k* maka batasan kecepatan yang diberikan juga semakin kecil sehingga perpindahan yang perlahan dan tidak terpaut jauh ini membuat partikel menemukan solusi terbaik yang sebenarnya masih belum solusi optimal dan masih membutuhkan iterasi yang lebih lanjut. Semakin besar nilai *k* juga akan membuat batasan kecepatan semakin besar sehingga dengan kecepatan yang lebih besar akan membuat perpindahan posisi yang lebih cepat, sehingga pada iterasi maksimal yaitu 10, partikel dianggap sudah menemukan solusi yang terbaik ditunjukkan nilai fitness yang lebih besar. Sehingga nilai pembatasan kecepatan yang terlalu besar atau terlalu kecil akan membuat partikel cepat memutuskan ke posisi solusi atau konvergensi dini.



Gambar 6.4 Grafik Konstanta Pembatasan Kecepatan

6.2.5 Hasil Pengujian terhadap banyaknya jumlah partikel

Pengujian jumlah partikel akan mempengaruhi dalam pencarian fitness optimal. Pengujian nilai konstanta pembatasan dilakukan dengan menggunakan data sebagai berikut:

1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.
3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN adalah 10000
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium
7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai w_{min} = 0.3 dan w_{max} = 0.6

11. Nilai c_1 = 1 dan c_2 =1.5

12. Nilai k = 0.6

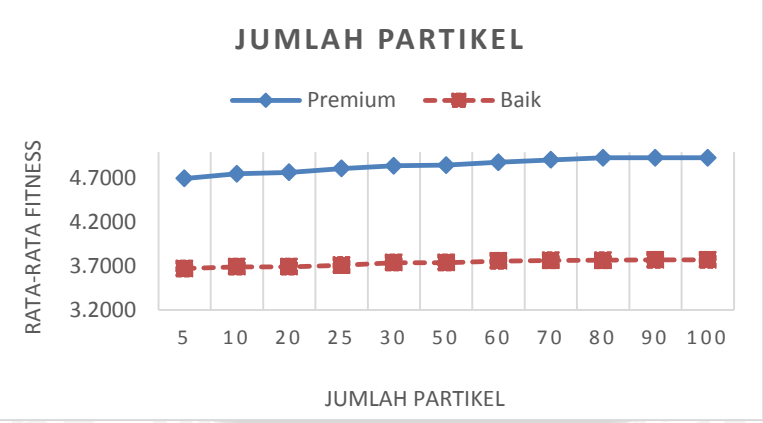
13. Jumlah iterasi maksimal PSO = 10 iterasi.

Hasil pengujian jumlah partikel untuk kualitas susu Premium ditunjukkan pada Tabel 6.5. Tabel hasil pengujian jumlah partikel untuk kualitas susu Premium dan Baik, secara lengkap akan dilampirkan pada Lampiran 7.

Tabel 6.5 Hasil Pengujian banyaknya jumlah partikel

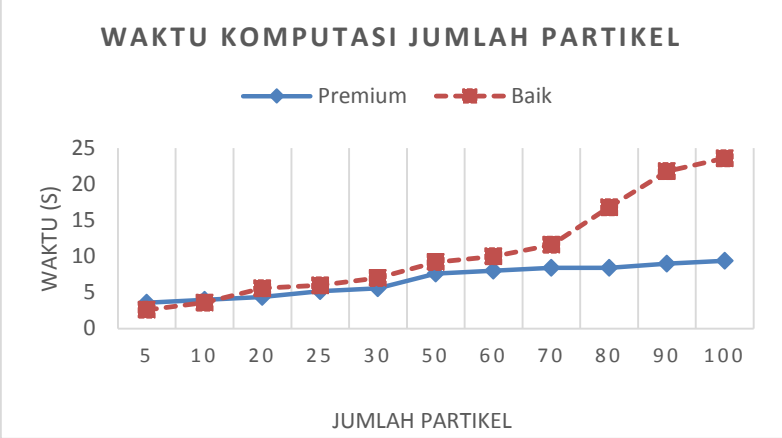
swarmsize	**Nilai fitness pada percobaan ke-*i***					**Rata-rata fitness**
	1	**2**	**3**	**4**	**5**	
5	4.1095	6.2948	4.0611	4.9568	4.0797	4.7004
10	4.1171	3.8609	3.9571	4.1415	7.6826	4.7518
20	3.8444	4.1252	4.6139	7.381	3.8769	4.7683
25	5.4044	4.3511	5.0925	4.474	4.7452	4.8134
30	3.6035	4.574	3.9775	7.9708	4.096	4.8444

Pengujian jumlah partikel digunakan untuk menganalisa jumlah partikel yang dapat menghasilkan fitness terbesar. Grafik pengujian berdasarkan hasil rata-rata fitness ditunjukkan pada Gambar 6.6. Berdasarkan grafik pengujian Gambar 6.5 rata-rata nilai fitness terbesar didapatkan dengan jumlah partikel sebanyak 30 partikel. Semakin banyak jumlah partikel yang diberikan maka akan semakin tinggi rata-rata fitness yang dihasilkan.



Gambar 6.5 Grafik Pengujian Jumlah Partikel

Grafik pengujian jumlah partikel berdasarkan dengan waktu komputasi ditunjukkan pada Gambar 6.6.



Gambar 6.6 Grafik waktu komputasi jumlah partikel

Semakin banyak jumlah partikel maka akan menyebabkan waktu komputasi setiap iterasi menjadi semakin lama (Engelbrecht A. P., 2007). Hal ini ditunjukkan pada Gambar 6.6 yang menunjukkan pengaruh jumlah partikel terhadap waktu komputasi. Berdasarkan Gambar 6.5 dan Gambar 6.6, direkomendasikan menggunakan jumlah partikel sebanyak 30 karena saat dilakukan, hasil rata-rata fitness menghasilkan fitness yang cukup besar dengan waktu komputasi yang kecil. Jumlah partikel tersebut berlaku pada kedua kualitas susu, yaitu Premium dan Baik.

6.2.6 Hasil Pengujian terhadap jumlah maksimum iterasi

Pengujian banyaknya jumlah maksimum iterasi yang dilakukan mempengaruhi pencarian nilai fitness yang optimal. Pengujian banyaknya maksimum iterasi dilakukan dengan menggunakan data sebagai berikut:

1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.
3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN adalah 10000
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium
7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai w_{min} = 0.3 dan w_{max} = 0.6
11. Nilai c_1 = 1 dan c_2 =1.5

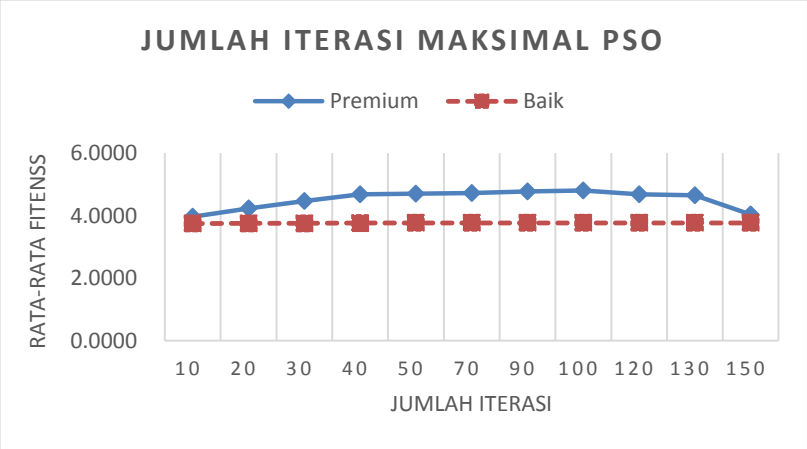
12. Nilai *k* = 0.6

13. Jumlah partikel = 30 partikel.

Hasil pengujian jumlah maksimum iterasi untuk kualitas susu Premium ditunjukkan pada Tabel 6.7, sedangkan untuk hasil pengujian kualitas susu Baik akan di lampirkan pada Lampiran 8. Grafik pengujian ditunjukkan pada Gambar 6.7.

Tabel 6.6 Hasil Pengujian jumlah maksimum iterasi

Maks. iterasi	**Nilai fitness pada percobaan ke-*i***					**Rata-rata fitness**
	1	**2**	**3**	**4**	**5**	
10	4.1154	4.0756	3.9312	3.9265	3.7492	3.9596
20	4.1321	4.4295	4.6131	3.9041	4.0695	4.2297
30	3.3419	4.1102	4.4426	4.0540	6.3731	4.4644
40	4.1060	4.1270	6.3219	4.1142	4.7442	4.6827
50	4.0742	5.7752	3.3787	4.2396	6.0288	4.6993
70	5.9157	4.1423	4.6401	4.5250	4.1645	4.6775
90	5.2161	4.1382	4.8562	4.0052	5.6343	4.7700
100	4.2148	5.0419	6.3328	4.2479	4.1976	4.8070
120	5.3183	4.0501	4.3021	3.4921	6.259	4.6843
130	6.3443	4.6457	4.0405	4.1497	4.088	4.6536
150	4.0061	3.7671	4.0592	4.2697	4.0736	4.0351



Gambar 6.7 Grafik Pengujian maksimum iterasi PSO

Hasil pengujian maksimal iterasi pada PSO yang ditunjukkan pada Tabel 6.6 divisualisasikan pada Gambar 6.7 dengan rata-rata nilai fitness terbesar

didapatkan dengan jumlah iterasi maksimal 100 iterasi yaitu menghasilkan fitness 4.807, sedangkan pada kualitas Baik, grafik juga meningkat walaupun kenaikan fitness hanya sedikit, hal ini dikarenakan batas bawah dan batas atas dari kandungan gizi susu kualitas Baik hanya memiliki selisih 0.2. Kualitas susu Baik pada iterasi 100 menghasilkan fitness sebesar 3.7812. Semakin besar nilai iterasi maksimal akan didapatkan nilai fitness yang semakin tinggi juga. Penentuan jumlah maksimum iterasi bergantung pada permasalahan yang dihadapi, sehingga pada penelitian ini, iterasi 100 dianggap cukup memberikan hasil fitness yang terbaik.

6.2.7 Hasil Pengujian terhadap kombinasi variasi *output* yang dioptimalkan

Pengujian variasi output dilakukan untuk mengetahui kualitas solusi yang didapatkan. Pada kasus pengujian ini merupakan untuk protein optimal dan lemak minimal. Pengujian ini dilakukan dengan menggunakan data sebagai berikut:

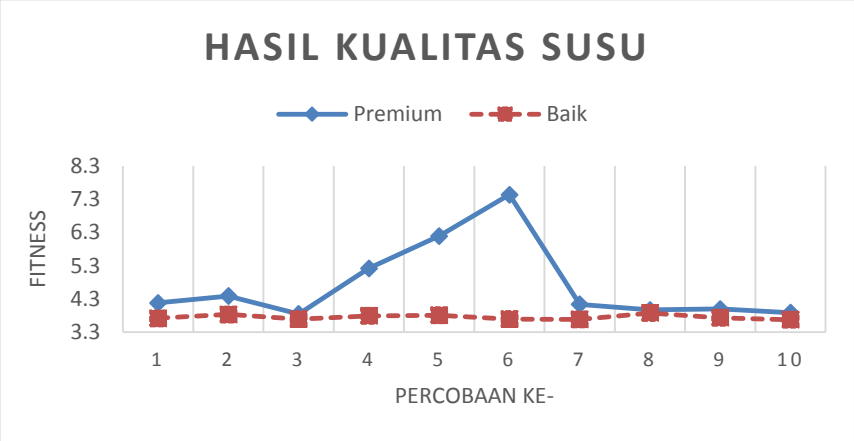
1. Jumlah data latih sama dengan data uji. Kualitas susu premiummemiliki data sebanyak 30 data, sedangkan kualitas Baik memiliki data sebanyak 15 data.
2. *Learning rate* yang digunakan adalah 0.64.
3. *Hidden layer* sejumlah 4.
4. Jumlah iterasi maksimal ANN adalah 10000
5. Berat badan kambing = 32 kg
6. Data Latih = 30 data Premium
7. Data uji = 30 data Premium
8. Kandungan gizi optimal = Protein
9. Kandungan gizi minimal = Lemak
10. Nilai w_{min} = 0.3 dan w_{max} = 0.6
11. Nilai c_1 = 1 dan c_2 =1.5
12. Nilai k = 0.6
13. Jumlah partikel = 30 partikel.
14. Iterasi maksimal PSO = 100 iterasi.

Tabel hasil pengujian variasi *output* untuk kualitas Premium ditunjukkan pada Tabel 6.7. Sedangkan untuk tabel hasil secara lengkap untuk kualitas susu Premium dan Baik serta variasi *output* yang lain akan ditunjukkan pada Lampiran 9.

Tabel 6.7 Hasil Pengujian Variasi *Output* kualitas Premium

Kasus Optimasi	Nilai fitness pada percobaan ke-*i*					Rata-rata fitness
	1	2	3	4	5	
Protein Optimal dan Lemak Minimal	4.18	4.388	3.8503	5.2234	6.1925	4.72588

Dari Tabel 6.7 didapatkan hasil rata-rata fitness untuk pengujian variasi output dalam mengoptimalkan Protein dan meminimalkan lemak sebesar 4.7258. Tabel 6.7 akan divisualisasikan dalam bentuk grafik pada Gambar 6.8. Sedangkan untuk hasil kenaikan dan penurunan output dapat dilihat pada Gambar 6.9

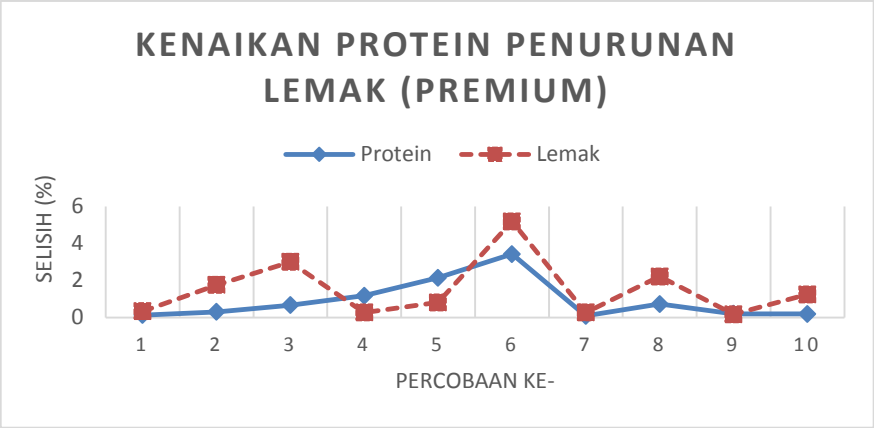


Gambar 6.8 Hasil variasi *output* Protein Optimal dan Lemak Minimal

Hasil kenaikan protein dan penurunan lemak untuk kualitas susu Premium ditunjukkan pada Tabel 6.8. Hasil pengujian secara lengkap untuk kualitas susu Premium dan Baik akan ditunjukkan pada Lampiran 10.

Tabel 6.8 Hasil pengujian kenaikan protein dan penurunan lemak

Kasus Optimasi Protein Optimal dan Lemak Minimal	Nilai fitness pada percobaan ke-*i*					Rata-rata (%)
	1	2	3	4	5	
Protein Naik	0.1283	0.3048	0.6669	1.1721	2.1471	0.904
Lemak turun	0.3389	1.7587	2.997	0.2708	0.81538	1.527



Gambar 6.9 Hasil Kenaikan Protein dan Penurunan Lemak

Dari Gambar 6.8 dapat dilihat bahwa dalam 10 percobaan dapat menghasilkan nilai fitness terbaik yaitu 7.4383 dengan kenaikan protein dan penurunan lemak yang ditunjukkan pada Gambar 6.9 yaitu pada percobaan ke-6 protein naik sebesar 3.4239% sedangkan lemak turun 5.19%.

BAB 7 PENUTUP

Pada bab penutup ini akan menjelaskan tentang kesimpulan dari penelitian ini serta saran yang diberikan peneliti untuk peneliti selanjutnya jika ingin melanjutkan penelitian dengan *object* dan metode yang sama.

7.1 Kesimpulan

1. Langkah pertama dalam implementasi algoritma Artificial Neural Network dan *Particle Swarm Optimization* adalah melakukan pelatihan data latih dan pengujian data uji. Data yang digunakan dalam pelatihan dan pengujian adalah kandungan nutrisi dari pakan kambing yang berupa Abu, Pk, Lk dan Sk sebagai data input. Sedangkan output kandungan gizi susu kambing yang diamati adalah Protein, lemak, Laktosa dan *Density*. Parameter ANN digunakan untuk melatih data dalam mencari bobot terbaik untuk menghasilkan nilai MSE terkecil. Bobot yang dihasilkan oleh pelatihan ANN akan digunakan dalam proses pengujian dan optimasi PSO. Semakin kecil nilai MSE yang dihasilkan dalam pelatihan, maka akan meningkatkan nilai fitness yang optimal dalam PSO.
2. Hasil pengujian yang dilakukan dalam penelitian ini meliputi pengujian parameter pelatihan ANN serta parameter yang ada di dalam PSO. Pengujian pelatihan ANN dengan parameter *learning rate* adalah 0.64, jumlah *hidden layer* adalah 4. Jumlah data latih sama dengan jumlah data uji, yaitu 30 data untuk kualitas susu Premium dan 15 data untuk kualitas susu Baik, menghasilkan nilai MSE terkecil yaitu 0.0088 pada iterasi ke-10000 dengan 30 data kualitas susu Premium, serta MSE 0.008 dengan 15 data kualitas susu Baik. Pada pengujian parameter PSO, parameter optimal yang didapatkan adalah nilai kombinasi inersia dengan w_{max} = 0.6 dan w_{min} = 0.3. Nilai konstanta kecepatan yaitu c_1 = 1 dan c_2 = 1.5. Nilai konstanta pembatasan kecepatan yaitu k = 0.6 serta banyaknya jumlah partikel yaitu 30. Iterasi maksimal PSO yaitu 100 iterasi. Dari hasil pengujian dengan parameter optimal PSO tersebut dihasilkan nilai rata-rata fitness sebesar 4.807.
3. Perbandingan hasil *output* dalam kasus pengujian yaitu mengoptimalkan protein dan meminimalkan lemak. Sebelum optimasi, kandungan protein dengan kualitas susu Premium untuk berat badan 32 adalah 3.944%. Setelah dioptimasi rata-rata kandungan protein naik 3.4239% sehingga menjadi 7.3679%. Kandungan lemak sebelum optimasi adalah 8.942%, sesudah optimasi rata-rata kandungan lemak turun 5.19% menjadi 3.637%.

7.2 Saran

1. Hasil penelitian dari implementasi Algoritma ANN dan PSO untuk memodelkan pakan serta mengoptimalkan kandungan gizi susu kambing ini dapat dikembangkan dan dijadikan bahan penelitian lebih lanjut tidak hanya dari segi mengoptimalkan kandungan gizi susu namun juga dapat meminimalkan harga pakan yang dikeluarkan serta harga jual yang diperoleh dengan kandungan gizi susu yang sudah dioptimalkan.
2. Pada penelitian ini terdapat masalah dalam pencarian jumlah iterasi ANN untuk mencapai konvergen, sehingga terdapat saran untuk mencoba algoritma *nguyen-widrow* untuk dijadikan sebagai perbandingan dalam pencarian konvergensi serta bobot yang didapatkan.

DAFTAR PUSTAKA

A. Ratnaweera, S. H. (2004). Self-organizing hierarchical particle. *IEEE Transactions on Evolutionary Computation 8*, 240–255.

Aliaga, I. M. (2003). Study of Nutritive Utilization of Protein and Magnesium in Rats With Resection FF The Distal Small Intestine. *Beneficial Effect of Goat Milk, J. Dairy Science*, vol 86: 2968-2966.

Atmiyati. (2001). Potensi Susu Kambing Sebagai Obat dan Sumber Protein Hewani Untuk Meningkatkan Gizi Petani. *Balai Penelitian Ternak*, Bogor.

Blakely, J. d. (1991). Ilmu Peternakan. *Terjemahan: B. Srigandono Gajah Mada University Press Yogyakarta*.

Cahyono, B. (2006). Beternak Domba dan Kambing. *Yogyakarta: Kanisius, Cetakan ke Sembilan*.

Chamidah, N., Wiharto, & Salamah, U. (2012). Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagasi Gradient Descent Adaptive Gain (BPGDAG) untuk Klasifikasi. *Jurnal ITSMART, Vol.1, No.1*, 28-33.

Chen, H.-L. e. (2011). A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest. *Knowledge-Based Systems 24*, 1348–1359.

Edo, R. Z. (2014). Implementasi *Backpropagation* Neural Network dalam Pembangkitan Otomatis Fungsi Keanggotaan Fuzzy pada Penderita Penyakit Hepatitis. *Skripsi Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya*.

Engelbrecht, A. P. (2007). Computational Intelligence: An Introduction 2nd ed. *West Sussex: John Wiley & Sons Ltd*.

Engelbrecht, A. P. (2007). *Computational Intelligence: An Introduction 2nd ed.* West Sussex: John Wiley & Sons Ltd.

Fadilah, A. N. (2015). Implementasi Analytical Hierarchy Process (AHP) dan Algoritma Genetika untuk Rekomendasi dan Optimasi Pemupukan Berimbang Tanaman Hortikultura. *Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 14*.

Fernanda, J. W. (2012). Boosting Neural Network dan Boosting Cart pada Klasifikasi Diabetes Militus Tiper II. *Jurnal Matematika*, Vol. 2, No. 2 , 33-49.

Fernanda, J. W., & Otok, B. W. (2012). Boosting Neural Network dan Boosting Cart pada Klasifikasi Diabetes Militus Tiper II. *Jurnal Matematika, Vol. 2, No. 2*, 33-49.

Gema, R. L. (2014). Jaringan Syaraf Tiruan Propagasi Balik dalam Prediksi Persediaan Ternak Sapi Potong (Studi Kasus di Wilayah Sumatera Barat). *Jurnal KomTekInfo, Fakultas Ilmu Komputer, Vol.1, No.2*, 21-27.

Gunawan, H. (2013). Prospek Usaha Penggemukan Kambing Potong. *Pustaka Baru Press*.

Hamed, H. N. (2006). *Particle Swarm Optimization* For Neural Network Learning Enhancement. *Malaysia*.

Hidayati, A. (2012). Pakan Kambing Perah. *Retrieved Januari 20, 2016, from Universitas Muhammadiyah Malang: http://peternakan.umm.ac.id/id/umm-news-2862-pakan-kambing-perah.html*.

Koten, B. W. (2014). Konsumsi Nutrien Ternak Kambing yang Mendapatkan Hijauan Hasil Tumpangsari Arbila (Phaseolus lunatus) dengan Sorgum sebagai Tanaman Sela pada Jarak Tanam Arbila dan Jumlah Baris Sorgum yang Berbeda. *Ilmu Ternak*, Vol.1, No.8, 38-45.

Mahmudy, W. (2015). Improved *Particle Swarm Optimization* untuk menyelesaikan permasalahan part type selection dan machine loading pada flexible manufacturing system (FMS). *Konferensi Nasional Sistem Informasi*.

Mahmudy, W. F. (2014.). Optimasi Part Type Selection And Machine Loading Problems Pada FMS Menggunakan Metode *Particle Swarm Optimization*. *Konferensi Nasional Sistem Informasi 2014.* Makassar.

Marini Putri Marwah, Y. Y. (2010). Produksi Dan Komposisi Susu Kambing Peranakan Ettawa Yang Diberi Suplemen Daun Katu (Sauropus androgynus (L.) Merr) Pada Awal Masa Laktasi . *Buletin Peternakan*, Vol. 34(2): 94-102.

Marini, F. &. (2015). *Particle Swarm Optimization* (PSO). A tutorial. *IEEE Chemometrics and Intelligent Laboratory Systems, 13.*

Matondang, Z. A. (2013). Jaringan Saraf Tiruan Dengan Algoritma *Backpropagation* Untuk Penentu Kelulusan Sidang Skripsi. *Pelita Informatika Budi Darma*, Volume : IV, Nomor: 1.

Mulyanto, S. (2006). Identifikasi Laktoferin Pada Kolostrum Dan Susu Kambing Dengan Metode Single Radial Immunodifusi Dan SDS-PAGE. *Fakultas Peternakan Institut Pertanian Bogor*.

Nugroho, B. A. (2010). Pasar Susu Dunia dan Posisi Indonesia.

Pramesti, D., Mahmudy, W. F., & Indriati. (2015). Optimasi Komposisi Pakan Kambing Potong Menggunakan Algoritma Genetika. *DORO : Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 5, No. 13*.

Prihatminingsih, G. E. (2015). Hubungan antara Konsumsi Protein dengan Produksi, Protein dan Laktosa Susu Kambing Peranakan Ettawa. *Jurnal Ilmu-ilmu Peternakan 25*, Vol.2, 20-27.

Purbayanto, A. T. (2009). Efek Pengaturan Suhu Outlet Pada Pengering Semprot Terhadap Sifat Fisik, Kimia, Dan Mikrobiologi Susu Kambing Bubuk. *Fakultas Teknologi Pertanian Institut Pertanian Bogor*.

Raharjo, J. S. (2013). Model Artificial Neural Network. *Jurnal Sistem Komputer*, Vol.3, No.1, ISSN: 2087-4685, e-ISSN: 2252-3456.

S.-C. Chu, R. J. (2004). Ant colony system with communication strategies. *Information Sciences*, vol.167. pp.63-76.

Shibata, K., & Ikeda, Y. (2009). Effect of Number of Hidden Neurons on Learning in Large-Scale Layered Neural Networks. *in Proccedings of the ICRPS-SICE International Joint Conference 2009 (ICCAS-SICE '09)*, pp. 5008-5013.

Subekti, E. (2009). Ketahanan Pakan Ternak Indonesia. *Mediagro*, Vol.5, No.2, 63-71.

Susilawati, T. &. (2011). Agribisnis Kambing. *Malang: UB Press, Cetakan Pertama*.

Yayu Zuriati, R. M. (2011). Karakteristik Kualitas Susu Segar Dan Yoghurt Dari Tiga Bangsa Kambing Perah Dalam Mendukung Program Ketahanan Dan Diversifikasi Pangan. *Seminar Nasional Teknologi Peternakan dan Veteriner*.

Zerda, E. R. (2009). Analisis dan Penerapan Algoritma *Particle Swarm Optimization* (PSO) pada Optimasi Penjadwalan Sumber Daya Proyek. *Bandung*.

LAMPIRAN 1 SURAT PERMOHONAN DATA SKRIPSI

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
Gedung A PTIIK Lt. 1, JL. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
http://ptiik.ub.ac.id E-mail : ptiik@ub.ac.id

Nomor : 1441 /UN10.36/AK/2106 24 FEB 2016
Perihal : ***Permohonan data skripsi***

Yth. Kepala Dinas Peternakan Provinsi Jawa Timur
Jl. Jend. A. Ayani 202
Fax (031) 8291853 - 8287165
email : disnak@jatimprov.go.id
Surabaya

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama	:	**Canny Amerilyse C**
NIM	:	**125150201111023**
Dosen P. A	:	**Indriati, S.T., M.Kom.**
Prodi	:	**Informatika / Ilmu Komputer**

Guna memperoleh data untuk skripsi mahasiswa tersebut di instansi saudara, jenis data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data	:	**Data tertulis mengenai data kandungan nutrisi bahan pakan kambing, data komposisi pakan untuk kambing serta data hasil laboratorium untuk kandungan gizi susu**
Waktu	:	**01 Maret 2016 - 04 Maret 2016**

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Mengetahui,
a.n. Ketua Program
Kepala Tata Usaha PTIIK,

Dra. Endang Budha Mahadyawati, M.AP.
NIP. 19581213 19820 2 2001

Dosen Pembimbing Akademik

Indriati, S.T., M.Kom.
NIK/ NIP. 200909 831013 2 001

Tembusan Kepada Yth:
1. Ketua Program Studi Informatika / Ilmu Komputer
2. UPTPT dan HMT Malang
3. Mahasiswa yang bersangkutan

LAMPIRAN 2 DATA KANDUNGAN GIZI SUSU

No.	Protein (%)	Fat (%)	Lactose (%)	Density	Bobot Badan (kg)	Klasifikasi Berdasarkan THAI
1	4.11	7.18	3.93	28.49	32	Premium
2	4.08	7.07	3.89	28.95	32	Premium
3	3.83	8.19	3.67	25.33	32	Premium
4	3.97	6.52	3.79	27.8	32	Premium
5	8.7	13.59	8.29	61.45	32	Premium
6	3.89	6.25	3.14	22.3	32	Premium
7	3.84	7.5	3.67	25.93	32	Premium
8	3.81	7.45	3.64	25.71	32	Premium
9	3.83	6.72	3.65	26.45	32	Premium
10	4.13	7.48	3.94	28.36	32	Premium
11	4.18	7.15	3.99	29.08	32	Premium
12	4.15	7.36	3.96	28.61	32	Premium
13	4.13	6.67	3.94	29.01	32	Premium
14	3.97	6.91	3.79	27.52	32	Premium
15	4	7.5	3.83	27.3	32	Premium
16	3.98	11.59	7.97	59.49	32	Premium
17	3.97	9.84	7.73	45.98	32	Premium
18	3.83	9.32	5.93	48.91	32	Premium
19	4.15	10.20	5.93	46.71	32	Premium
20	3.79	7.91	6.06	50.76	32	Premium
21	4.09	9.64	3.65	23.46	32	Premium
22	3.88	10.91	5.21	49.22	32	Premium
23	3.98	7.24	6.07	35.32	32	Premium
24	4.02	9.75	6.98	51.93	32	Premium
25	3.8	11.21	5.05	28.95	32	Premium
26	3.83	6.33	5.24	47.57	32	Premium
27	3.97	10.97	4.43	48.78	32	Premium
28	4.04	8.57	7.13	28.59	32	Premium
29	3.84	7.91	5.72	43.93	32	Premium
30	3.87	6.98	4.58	44.01	32	Premium
31	3.63	8	3.48	23.8	32	Baik
32	3.56	6.71	3.41	24.28	32	Baik
33	3.53	8.39	3.38	22.64	32	Baik
34	3.48	7.13	5.21	47.07	32	Baik
35	3.52	12.81	7.76	58.57	32	Baik
36	3.64	10.64	6.33	40.79	32	Baik
37	3.53	6.5	5.07	31.13	32	Baik
38	3.64	13.20	7.57	37.23	32	Baik

39	3.68	12.57	4.93	37.96	32	Baik
40	3.58	11.42	3.98	50.73	32	Baik
41	3.67	12.14	3.64	37.36	32	Baik
42	3.52	9.77	4.2	37.39	32	Baik
43	3.65	7.67	5	30.06	32	Baik
44	3.56	12.74	6.98	45.24	32	Baik
45	3.63	10.81	4.71	35.71	32	Baik
46	3.95	7.76	3.78	26.64	34	Premium
47	3.88	5.71	3.7	27.76	34	Premium
48	3.99	7.36	3.82	27.33	34	Premium
49	3.81	7.38	3.65	25.82	34	Premium
50	4.09	9.91	3.92	26.05	34	Premium
51	3.71	7.69	3.54	24.66	34	Premium
52	4.09	6.89	3.91	28.53	34	Premium
53	3.97	7.18	3.79	27.3	34	Premium
54	3.88	6.29	3.71	27.29	34	Premium
55	4.05	6.43	3.87	28.59	34	Premium
56	4.06	7.3	3.87	27.9	34	Premium
57	3.93	7.31	3.75	26.81	34	Premium
58	4.09	12.82	3.93	23.67	34	Premium
59	3.95	6.95	3.77	27.26	34	Premium
60	3.79	6.38	3.62	26.42	34	Premium
61	3.78	8.65	3.67	28.21	34	Premium
62	3.72	8.66	3.9	24.28	34	Premium
63	3.71	12.07	3.53	27.85	34	Premium
64	3.78	9.78	3.78	28.26	34	Premium
65	3.83	7.06	3.6	28.5	34	Premium
66	3.86	5.9	3.56	24.15	34	Premium
67	3.98	11.01	3.53	23.96	34	Premium
68	4.05	11.33	3.81	25.9	34	Premium
69	3.8	12.35	3.76	23.9	34	Premium
70	3.77	8.67	3.67	26.46	34	Premium
71	3.83	7.16	3.61	24.67	34	Premium
72	3.97	7.99	3.58	26.63	34	Premium
73	3.87	7.28	3.41	27.45	34	Premium
74	4.04	8.54	3.81	26.65	34	Premium
75	3.8	6.97	3.48	27.53	34	Premium
76	3.6	8.11	3.44	23.41	34	Baik
77	3.58	7.94	3.43	23.42	34	Baik
78	3.55	6.71	3.39	24.14	34	Baik
79	3.67	8.45	3.8	28.36	34	Baik
80	3.55	11.61	3.89	27.9	34	Baik
81	3.69	8	3.43	23.46	34	Baik

82	3.64	5.97	3.91	28.4	34	Baik
83	3.45	8.66	3.81	27.16	34	Baik
84	3.53	5.86	3.72	24.32	34	Baik
85	3.63	8.95	3.39	25.12	34	Baik
86	3.69	5.78	3.78	28.27	34	Baik
87	3.57	11.22	3.88	25.13	34	Baik
88	3.54	6.63	3.93	26.61	34	Baik
89	3.63	6.51	3.67	28.24	34	Baik
90	3.69	8.83	3.67	26.52	34	Baik
91	3.92	8.39	3.75	25.89	36	Premium
92	3.81	7.05	3.64	26.04	36	Premium
93	4.54	7.03	4.33	32.16	36	Premium
94	3.87	6.2	3.69	27.25	36	Premium
95	3.71	8.28	3.55	24.19	36	Premium
96	3.84	6.77	3.66	26.5	36	Premium
97	3.85	6.7	3.68	26.69	36	Premium
98	3.92	6.57	3.74	27.32	36	Premium
99	3.87	7.49	3.7	26.2	36	Premium
100	3.79	9.09	3.63	24.2	36	Premium
101	4.24	7.49	4.05	29.25	36	Premium
102	4.2	8.01	4.01	28.5	36	Premium
103	4.29	7.32	4.09	29.81	36	Premium
104	4.21	7.33	4.02	29.18	36	Premium
105	4.1	7.68	3.91	27.92	36	Premium
106	4.05	12	3.89	24.02	36	Premium
107	4.16	7.22	3.97	28.81	36	Premium
108	4.14	7.57	3.95	28.37	36	Premium
109	4.13	10.85	3.75	31.39	36	Premium
110	3.87	6.41	3.55	24.67	36	Premium
111	4.16	8.95	3.88	28.31	36	Premium
112	4.29	9.42	4.18	30.33	36	Premium
113	3.87	10.56	4.19	25.16	36	Premium
114	3.82	8.85	3.92	26.6	36	Premium
115	4.1	7.33	4	26.56	36	Premium
116	3.79	9.23	3.83	26.9	36	Premium
117	3.733	6.59	4.12	30.63	36	Premium
118	3.747	8.31	4.19	28.35	36	Premium
119	3.72	11.36	4.09	30.85	36	Premium
120	3.78	9.75	4.12	27.67	36	Premium
121	3.63	7.16	3.87	24.22	36	Baik
122	3.58	6.95	3.68	28.24	36	Baik
123	3.42	11.21	3.59	27.05	36	Baik
124	3.45	9.83	3.82	27.72	36	Baik

125	3.42	9.61	4.13	28.68	36	Baik
126	3.41	6.98	4.09	24.8	36	Baik
127	3.5	9.59	4.17	29.15	36	Baik
128	3.49	9.87	4.21	30.69	36	Baik
129	3.52	7.21	3.96	32.05	36	Baik
130	3.49	9.1	4.17	28.18	36	Baik
131	3.52	8.02	4.24	27.83	36	Baik
132	3.66	6.51	3.83	31.67	36	Baik
133	3.61	6.64	3.65	26.73	36	Baik
134	3.68	6.32	3.68	26.58	36	Baik
135	3.56	8.49	4.32	28.5	36	Baik
136	4.32	7.48	4.12	29.93	38	Premium
137	3.91	6.87	3.73	27.01	38	Premium
138	3.83	5.95	3.66	27.15	38	Premium
139	3.78	7.93	3.62	25.12	38	Premium
140	3.75	8.33	3.59	24.52	38	Premium
141	3.97	7.26	3.8	27.25	38	Premium
142	4.3	6.19	4.1	30.82	38	Premium
143	3.82	7.27	3.65	25.94	38	Premium
144	3.78	7.35	3.62	25.59	38	Premium
145	3.74	5.72	3.57	26.59	38	Premium
146	4.13	6.13	3.94	29.45	38	Premium
147	4.1	6.03	3.91	29.32	38	Premium
148	4.18	6.47	3.99	29.64	38	Premium
149	4.28	6.87	4.08	30.09	38	Premium
150	4.25	7.9	4.06	29.03	38	Premium
151	4.2	11.83	4.04	25.43	38	Premium
152	3.96	7.96	3.79	26.56	38	Premium
153	4.18	5.99	3.99	30.02	38	Premium
154	3.785	6.17	3.89	28.08	38	Premium
155	3.749	9.04	3.63	27.25	38	Premium
156	3.794	10.58	3.63	24.66	38	Premium
157	3.76	9.42	3.73	26.51	38	Premium
158	3.76	9.66	3.66	25.15	38	Premium
159	3.88	6.1	3.93	25.37	38	Premium
160	4.1	6.62	3.94	28.26	38	Premium
161	3.96	6.76	3.95	29.02	38	Premium
162	3.74	10.12	4.08	29.56	38	Premium
163	3.86	8.55	3.58	26.82	38	Premium
164	4.03	8.24	4	28.86	38	Premium
165	3.73	7.69	4.04	28.79	38	Premium
166	3.41	9.61	3.76	29.46	38	Baik
167	3.45	8.81	3.94	26.56	38	Baik

168	3.56	11.59	3.63	28.56	38	Baik
169	3.65	10.51	4.11	26.45	38	Baik
170	3.68	10.28	3.59	26.15	38	Baik
171	3.41	10.26	3.61	29.85	38	Baik
172	3.42	8.54	3.78	29.87	38	Baik
173	3.54	11.75	3.91	25.75	38	Baik
174	3.56	11.05	4.06	25.77	38	Baik
175	3.44	6.04	3.74	30.19	38	Baik
176	3.42	8.88	3.74	26.36	38	Baik
177	3.65	7.66	3.73	30.6	38	Baik
178	3.69	9.33	3.76	29.98	38	Baik
179	3.57	8.05	3.7	28.28	38	Baik
180	3.55	5.78	4.03	28.88	38	Baik
181	3.95	7.49	3.77	26.86	40	Premium
182	3.95	6.45	8.77	27.69	40	Premium
183	3.94	6.51	3.76	27.55	40	Premium
184	3.91	9.26	3.75	26.09	40	Premium
185	3.93	6.83	3.75	27.24	40	Premium
186	3.9	4.93	3.71	28.51	40	Premium
187	3.77	7.5	3.6	25.35	40	Premium
188	3.79	7.4	3.62	25.58	40	Premium
189	3.86	7.54	3.69	26.11	40	Premium
190	3.76	7.82	3.6	25.04	40	Premium
191	3.84	6.79	3.67	26.51	40	Premium
192	4.03	6.5	3.84	28.31	40	Premium
193	3.97	6.77	3.78	27.57	40	Premium
194	4.2	5.74	4	30.39	40	Premium
195	4.06	11.67	3.9	24.34	40	Premium
196	4.12	7.53	3.94	28.25	40	Premium
197	4.04	5.85	3.85	28.92	40	Premium
198	3.87	5.79	8.18	23.13	40	Premium
199	3.96	9.2	8.28	26.6	40	Premium
200	4.05	6.58	5.31	23.92	40	Premium
201	3.78	11.31	8.55	28.8	40	Premium
202	4.12	11.19	5.83	24.04	40	Premium
203	3.74	9.89	8.36	27.59	40	Premium
204	3.76	9.3	6.45	25.21	40	Premium
205	3.91	9.28	6.40	27.18	40	Premium
206	3.754	4.96	8.47	26.02	40	Premium
207	3.809	8.19	5.17	29.26	40	Premium
208	3.776	10.83	3.39	28.09	40	Premium
209	4.074	8.47	4.34	24.63	40	Premium
210	4.099	6.99	7.31	28.35	40	Premium

211	3.41	6.72	3.26	22.96	40	Baik
212	3.41	5.38	7.76	28.93	40	Baik
213	3.62	10	3.5	23.95	40	Baik
214	3.63	9.13	4.17	25	40	Baik
215	3.64	5.31	7.4	25.79	40	Baik
216	3.45	10.94	5.94	23.61	40	Baik
217	3.65	5.72	7.46	28.73	40	Baik
218	3.48	5.95	3.99	23.73	40	Baik
219	3.64	7.07	5.48	27.61	40	Baik
220	3.69	8.4	7.46	23.9	40	Baik
221	3.492	6.91	4.18	26.02	40	Baik
222	3.486	7.84	6.85	29.38	40	Baik
223	3.6	10.12	7.49	24.23	40	Baik
224	3.625	7.93	6.89	24.67	40	Baik
225	3.462	5.93	5.23	29.76	40	Baik

LAMPIRAN 3 HASIL WAWANCARA PAKAR

23 Desember 2015

1. Apa saja jenis kambing yang ada di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang?
   Jawab: Kambing perah peranakan etawa (PE) jenis kaligesing dan senduro.
   Narasumber: Ibu drh. Jaya Wulandari

2. Bagaimana ciri-ciri kambing PE yang dapat diambil susunya?
   Jawab: Kambing yang sedang dalam masa laktasi, yaitu kambing yang sudah pernah melahirkan dan sedang tidak hamil.
   Narasumber: Bapak Rokim, S.Pt dan Bapak Kayat

3. Berapa kali pemerahan susu yang dilakukan setiap satu harinya?
   Jawab: Kambing PE diperah setiap pagi hari jam 08.00 WIB dimana setiap satu hari proses pemerahannya hanya dilakukan satu kali saja.
   Narasumber: Bapak Rokim, S.Pt dan Bapak Kayat

4. Berapa jenis bobot badan kambing PE yang sudah siap perah?
   Jawab : Ada bermacam-macam jenis bobot badan kambing PE yang sudah siap perah, diantaranya 32 kg, 34 kg, 36 kg, 38 kg dan 40kg.
   Narasumber: Bapak Rokim, S.Pt dan Bapak Kayat

5. Apa saja jenis-jenis pakan hijauan dan leguminose yang tersedia di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang?
   Jawab : Ada 10 jenis pakan hijauan dan leguminose yang tersedia, yaitu rumput taiwan, rumput odot, rumput setaria, rumput raja, rumput *brachiaria decumbens*, kaliandra, gamal, lamtoro, turi dan tarum.
   Narasumber: Bapak Rokim, S.Pt

6. Diantara 10 jenis pakan hijauan dan leguminose yang tersedia, bahan pakan manakah yang sering diberikan pada kambing PE?
   Jawab: Rumput odot dan rumput raja.
   Narasumber: Bapak Rokim, S.Pt

7. Berapa kilogram pemberian pakan untuk kambing PE dalam satu harinya?
   Jawab: Pemberian pakan pada kambing PE diberikan berdasarkan bobot badan kambing tersebut. Pakan yang diberikan pada kambing PE perharinya, yaitu minimal 10% dari bobot badan kambing dan maksimal 12% dari bobot badan kambing.
   Narasumber: Bapak Rokim, S.Pt

8. Selain jenis pakan hijauan dan *leguminose*, adakah jenis pakan lainnya yang diberikan pada kambing PE?
   Jawab: Iya ada, selain pakan hijauan dan *leguminose*, kambing PE juga diberikan pakan berupa konsentrat yang berfungsi untuk tambahan asupan nutrisi pakan yang tidak diperoleh dari pakan hijauan dan *leguminose*.
   Narasumber: Bapak Rokim, S.Pt

9. Apa saja kandungan nutrisi yang terdapat pada pakan hijauan dan *leguminose*?
   Jawab: Kandungan nutrisi yang terdapat pada pakan sebenarnya banyak jenisnya, akan tetapi kandungan nutrisi dari pakan yang berpengaruh terhadap kualitas susu yang dihasilkan adalah abu, protein kasar, lemak kasar dan serat kasar.
   Narasumber: Bapak Rokim, S.Pt

10. Pernahkah pihak UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang memberikan komposisi pakan yang berbeda-beda setiap bulannya?
    Jawab : Pihak UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang tidak pernah memberikan treatment berupa pemberian komposisi pakan yang berbeda-beda setiap bulannya karena tidak semua kambing mampu beradaptasi terhadap makanan yang baru dan jika tidak cocok pada pencernaan kambing, maka akan berakibat buruk terhadap kesehatan kambing dan pemulihan kesehatan kambing bisa berlangsung cukup lama.
    Narasumber: Bapak Rokim, S.Pt

11. Adakah uji kandungan gizi susu kambing PE di UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang?
    Jawab: Ada, uji kandungan gizi susu kambing PE dilakukan di laboratorium milik UPT Pembibitan Ternak dan Hijauan Makanan Ternak Singosari-Malang menggunakan alat yang bernama lactoscan.
    Narasumber: Ibu drh. Jaya Wulandari dan Bapak Kayat

12. Apa saja kandungan gizi susu yang dihasilkan dari uji laboratorium?
    Jawab: Hasil dari uji laboratorium menggunakan lactoscan, yaitu ada lemak, SNF, *density*, *lactose*, *solids*, protein, *added water*, *temperature sample* dan *freeze point*.
    Narasumber: Ibu drh. Jaya Wulandari dan Bapak Kayat

13. Dari macam-macam kandungan gizi susu yang sudah dijelaskan sebelumnya, manakah kandungan gizi susu yang mempunyai pengaruh besar, baik dalam hal kesehatan ataupun kualitas susu yang disukai konsumen dan apa alasannya?
    Jawab: Kandungan gizi susu yang mempunyai pengaruh besar bagi kesehatan ataupun kualitas susu yang disukai konsumen adalah protein, lemak, laktosa dan *density*. Semakin tinggi kandungan protein, maka susu tersebut akan

semakin bagus untuk tubuh. Lemak susu juga bagus untuk tubuh, namun banyak orang yang alergi terhadap lemak susu sapi, sehingga susu kambing yang rendah lemak dapat dijadikan sebagai alternatif untuk orang-orang yang alergi pada lemak susu sapi. Besarnya *density* yang ada pada susu kambing akan membuat susu menjadi semakin kental dan semakin mahal harga jualnya. Selain itu, laktosa yang besar juga akan mempengaruhi rasa kombinasi manis dan asin pada susu.

Narasumber: Ibu drh. Jaya Wulandari

31 Maret 2016

1. Adakah klasifikasi hasil susu berdasarkan kandungan gizi susu yang diperoleh?
   Jawab: Ada. Terdapat tiga klasifikasi penggolongan kualitas susu, yaitu kelas premium, baik dan standart.
   Narasumber: Ibu drh. Jaya Wulandari

2. Bagaimana cara mengklasifikasikan penggolongan kualitas susu berdasarkan kelasnya?
   Jawab: Kandungan gizi susu yang dijadikan patokan adalah protein. Jika protein yang dihasilkan lebih dari 3.7%, maka susu tersebut digolongkan sebagai kualitas premium. Jika protein yang dihasilkan antara 3.4% - 3.7%, maka susu tersebut digolongkan sebagai kualitas baik dan jika protein yang dihasilkan kurang dari 3.4%, maka susu tersebut digolongkan sebagai kualitas standart.
   Narasumber: Ibu drh. Jaya Wulandari

3. Menurut Anda, kandungan gizi susu manakah yang dapat dioptimalkan dan diminimalkan?
   Jawab: Semakin tinggi kandungan protein yang ada di dalam susu, maka semakin bagus pula untuk kesehatan tubuh manusia sehingga protein dapat dioptimalkan kandungannya. Lemak pada susu kambing memiliki keistimewaan dibandingkan lemak susu sapi, yaitu kandungan lemak yang lebih rendah dari susu sapi sehinga lemak dapat diminimalkan kandungannya. Laktosa digunakan untuk menambah rasa pada susu kambing, jadi semakin tinggi kandungan laktosa yang ada pada susu, maka rasanya akan menjadi semakin gurih sehingga laktosa dapat dioptimalkan. Semakin tinggi kandungan density, maka susu yang dihasilkan juga akan semakin kental sehingga density dapat dioptimalkan. Susu yang memiliki rasa gurih dan kenatal adalah susu yang sangat disukai oleh konsumen.
   Narasumber: Ibu drh. Jaya Wulandari

4. Apakah ada formulasi baku yang menunjukkan pengaruh pakan yang diberikan terhadap hasil produksi susu kambing?
   Jawab: Untuk formulasi bakunya tidak ada, dari Dinas Provinsi juga tidak memberikan aturan baku mengenai pengaruh pakan kambing terhadap hasil susu, hanya saja sesuai dengan fakta di lapangan, jika pakan yang diberikan

memiliki kandungan nutrisi yang baik, juga pasti akan berpengaruh besar ke hasil susunya. Susu yang dihasilkan nanti juga akan memiliki kandungan gizi yang tinggi juga. Dan itupun tidak ada catatan tertulisnya, jadi berdasarkan pengalaman saja.

Narasumber: Bapak Rokim, S.Pt dan Ibu drh. Jaya Wulandari

LAMPIRAN 4 HASIL PENGUJIAN MAKSIMAL ITERASI ANN

Pengujian untuk susu Kualitas Premium

Learning rate = 0.64 dan 30 data Premium

Jumlah Iterasi	Nilai fitness pada percobaan ke-					Rata-rata MSE
	1	2	3	4	5	
10	0.0271	0.0291	0.03	0.029	0.0289	0.02882
20	0.0267	0.0286	0.0295	0.0285	0.0283	0.02832
50	0.0261	0.0283	0.0289	0.0274	0.0269	0.02752
100	0.0255	0.0281	0.0286	0.0264	0.0252	0.02676
150	0.0247	0.0279	0.0282	0.0256	0.0238	0.02604
200	0.0238	0.0277	0.0279	0.0249	0.0229	0.02544
400	0.0209	0.0263	0.0258	0.0223	0.0222	0.0235
600	0.015	0.023	0.0245	0.0218	0.0219	0.02124
800	0.012	0.022	0.024	0.0214	0.0207	0.02002
1000	0.0106	0.0213	0.0235	0.021	0.0193	0.01914
5000	0.0027	0.0097	0.017	0.0123	0.0087	0.01008
10000	0.0022	0.0091	0.0164	0.0087	0.0076	0.0088

Pengujian untuk susu Kualitas Baik

Learning rate = 0.64 dan 15 data Baik

Jumlah Iterasi	Nilai fitness pada percobaan ke-					Rata-rata MSE
	1	2	3	4	5	
10	0.033	0.0323	0.0326	0.032	0.0334	0.03266
20	0.0319	0.0305	0.032	0.0298	0.032	0.03124
50	0.0281	0.0274	0.0304	0.0243	0.0278	0.0276
100	0.0236	0.0269	0.0275	0.0216	0.0248	0.02488
150	0.0207	0.0269	0.0196	0.0197	0.0218	0.02174
200	0.0195	0.0267	0.0161	0.0183	0.0189	0.0199
400	0.0182	0.0245	0.0149	0.0159	0.0116	0.01702
600	0.0156	0.0218	0.014	0.0134	0.0077	0.0145

800	0.0099	0.0202	0.0129	0.0121	0.0065	0.01232
1000	0.0077	0.0195	0.0114	0.0118	0.006	0.01128
5000	0.0028	0.0162	0.0104	0.0068	0.0044	0.00812
10000	0.0031	0.0157	0.0128	0.0045	0.0039	0.008

LAMPIRAN 5 HASIL PENGUJIAN KOMBINASI NILAI INERSIA

Hasil percobaan dengan data Premium

w_{min}	w_{max}	Nilai fitness pada percobaan ke-										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	
0.2	0.6	3.928	3.886	4.585	4.021	7.950	4.022	4.102	6.330	4.195	4.098	4.711
	0.7	6.943	3.619	8.002	5.833	4.168	4.089	3.580	3.443	3.652	4.081	4.741
	0.8	3.734	4.069	6.168	3.984	3.572	4.950	4.226	4.715	4.526	4.908	4.485
0.3	0.6	4.731	4.322	4.168	7.132	6.343	6.120	4.200	4.178	3.386	4.768	4.935
	0.7	4.235	4.039	6.231	4.243	3.987	3.551	6.345	4.420	5.949	5.487	4.849
	08	4.688	8.469	4.262	4.967	5.296	3.966	4.288	4.586	4.124	3.858	4.850
0.4	0.6	5.397	3.913	5.449	7.454	4.141	3.815	4.128	4.034	4.276	4.155	4.676
	0.7	4.225	5.505	4.969	5.798	3.990	3.421	6.293	5.493	4.201	3.386	4.728
	0.8	3.416	4.867	4.553	7.662	4.092	7.683	4.226	3.684	3.724	3.335	4.724

Hasil percobaan dengan data Baik

w_{min}	w_{max}	Nilai fitness pada percobaan ke-										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	
0.2	0.6	3.644	3.682	3.682	3.775	3.769	3.747	3.683	3.793	3.838	3.644	3.361
	0.7	3.699	3.711	3.696	3.695	3.646	3.765	3.682	3.665	3.693	3.699	3.325
	0.8	3.777	3.865	3.721	3.695	3.743	3.694	3.739	3.703	3.741	3.777	3.368
0.3	0.6	3.775	3.772	3.781	3.732	3.771	3.760	3.798	3.685	3.693	3.775	3.377
	0.7	3.699	3.683	3.625	3.770	3.716	3.686	3.735	3.688	3.797	3.699	3.340
	08	3.770	3.661	3.771	3.682	3.684	3.757	3.656	3.682	3.783	3.770	3.344
0.4	0.6	3.725	3.775	3.706	3.807	3.772	3.751	3.738	3.689	3.792	3.725	3.376
	0.7	3.753	3.630	3.771	3.670	3.744	3.795	3.547	3.857	3.704	3.753	3.347
	0.8	3.683	3.706	3.776	3.682	3.714	3.682	3.688	3.688	3.752	3.683	3.337

LAMPIRAN 6 HASIL PENGUJIAN KONSTANTA AKSELERASI

Hasil percobaan dengan data Premium

c_1	c_2	Nilai fitness pada percobaan ke-										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	
1	1	4.3727	6.3504	3.6277	3.9439	3.8144	4.2473	3.7542	3.8413	3.992	5.174	4.3118
	1.5	4.5981	3.831	5.9101	4.1209	4.256	4.5618	6.272	3.5193	3.649	3.969	4.4687
	2	4.8374	4.1436	4.096	4.0405	4.7421	4.6595	4.0312	4.428	5.399	3.389	4.3765
1.5	1	4.1157	4.5687	4.056	4.6167	4.0892	4.3198	4.0673	4.1391	3.908	4.296	4.2176
	1.5	4.0358	4.0449	3.3778	4.2673	3.781	3.4309	4.5384	5.0625	4.041	4.036	4.0616
	2	3.4355	3.8107	4.2188	4.2229	6.6468	4.069	5.1781	4.0984	4.001	3.903	4.3584
2	1	4.0122	4.1035	4.1605	3.5717	6.1426	6.2995	4.0843	3.9555	4.095	3.985	4.4409
	1.5	5.3096	3.566	3.597	4.3381	6.4134	4.3303	4.0194	4.2154	3.548	3.936	4.3274
	2	3.338	5.8867	3.9761	4.3407	4.1932	5.1982	4.1436	4.2512	4.974	4.133	4.4434

Hasil percobaan dengan data Baik

c_1	c_2	Nilai fitness pada percobaan ke-										Rata-rata fitness
		1	2	3	4	5	6	7	8	9	10	
1	1	3.771	3.715	3.699	3.703	3.807	3.657	3.634	3.731	3.614	3.715	3.7045
	1.5	3.752	3.726	3.787	3.779	3.733	3.742	3.738	3.680	3.738	3.534	3.7209
	2	3.663	3.731	3.584	3.682	3.686	3.791	3.723	3.729	3.767	3.691	3.7047
1.5	1	3.651	3.700	3.686	3.746	3.663	3.729	3.695	3.691	3.682	3.726	3.6967
	1.5	3.784	3.662	3.680	3.728	3.700	3.742	3.732	3.682	3.606	3.839	3.7154
	2	3.681	3.682	3.649	3.642	3.661	3.680	3.747	3.790	3.874	3.708	3.7113
2	1	3.681	3.773	3.682	3.652	3.771	3.659	3.682	3.740	3.619	3.803	3.7060
	1.5	3.765	3.667	3.691	3.631	3.682	3.635	3.776	3.783	3.617	3.804	3.7050
	2	3.623	3.672	3.677	3.682	3.699	3.605	3.814	3.682	3.693	3.772	3.6918

LAMPIRAN 7 HASIL PENGUJIAN KONSTANTA PEMBATASAN KECEPATAN

Hasil percobaan dengan data Premium

k	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
0.1	4.2179	3.8941	3.5788	3.6121	6.8143	5.1913	3.9402	3.9847	4.0923	5.6999	4.50256
0.2	4.0627	5.8315	4.7374	4.0132	3.6645	4.2167	3.7707	4.0834	6.3194	6.7555	4.7455
0.3	4.9393	6.2623	3.3586	4.4449	4.0442	3.9769	4.11	4.2717	5.7844	4.126	4.53183
0.4	4.8989	3.689	3.8247	3.7322	3.983	4.8716	6.3263	6.1655	4.16	4.0065	4.56577
0.5	4.0612	7.7108	3.9321	4.1066	3.645	3.3976	3.7813	4.9327	4.033	3.7007	4.3301
0.6	3.338	5.8867	3.9761	6.3407	4.1932	5.1982	4.1436	4.2512	4.9735	4.133	4.64342
0.7	4.0814	3.782	4.0023	3.8355	3.7548	6.294	4.1181	4.9988	4.3564	6.585	4.58083
0.8	4.0287	3.9241	4.4896	6.3355	3.5075	4.25	3.6792	3.9076	4.1251	4.0926	4.23399
0.9	4.1379	6.3442	4.189	4.1587	4.0008	4.0841	3.9588	4.4859	4.4002	6.3346	4.60942

Hasil percobaan dengan data Baik

k	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
0.1	3.6815	3.7656	3.6859	3.537	3.7178	3.6731	3.6797	3.7651	3.7739	3.7611	3.70407
0.2	3.7455	3.7415	3.6591	3.7741	3.7307	3.6816	3.6815	3.8778	3.6398	3.629	3.71606
0.3	3.6501	3.5788	3.8345	3.5538	3.67	3.5783	3.6167	3.6834	3.6818	3.6862	3.65336
0.4	3.6815	3.6816	3.6864	3.6817	3.7037	3.7834	3.7052	3.6241	3.7487	3.6351	3.69314
0.5	3.7769	3.7773	3.6544	3.7455	3.7085	3.7871	3.7483	3.7489	3.72	3.6815	3.73484
0.6	3.8228	3.7723	3.7768	3.6815	3.6988	3.8047	3.8143	3.682	3.6932	3.7715	3.75179
0.7	3.7172	3.7502	3.7735	3.6557	3.7002	3.7039	3.7523	3.7227	3.6815	3.6388	3.7096
0.8	3.6615	3.7762	3.6815	3.7409	3.6821	3.7513	3.7639	3.6975	3.6034	3.7735	3.71318
0.9	3.7877	3.6624	3.6946	3.7341	3.7455	3.7434	3.6552	3.6815	3.7096	3.7568	3.71708

LAMPIRAN 8 HASIL PENGUJIAN JUMLAH PARTIKEL

Hasil percobaan dengan data Premium

swarmsize	Nilai fitness pada percobaan ke-					Rata-rata fitness
	1	2	3	4	5	
5	4.1095	6.2948	4.0611	4.9568	4.0797	4.7004
10	4.1171	3.8609	3.9571	4.1415	7.6826	4.7518
20	3.8444	4.1252	4.6139	7.381	3.8769	4.7683
25	5.4044	4.3511	5.0925	4.474	4.7452	4.8134
30	3.6035	4.574	3.9775	7.9708	4.096	4.8444
50	4.0739	6.6381	4.2228	5.0418	4.2803	4.8514
60	4.3275	6.6752	4.2437	4.1874	4.9844	4.8836
70	5.3631	3.4947	5.3571	6.075	4.271	4.9122
80	3.6214	5.8802	4.363	6.2552	4.5538	4.9347
90	3.7906	3.7993	6.4515	6.3605	4.2715	4.9347
100	4.7633	4.7305	4.8587	6.1433	4.1838	4.9359

Swarmsize	Waktu Komputasi Data Premium (s)
5	3.6
10	4
20	4.4
25	5.2
30	5.6
50	7.6
60	8
70	8.4
80	8.4
90	9
100	9.4

Hasil percobaan dengan data Baik

swarmsize	Nilai fitness pada percobaan ke-					Rata-rata fitness
	1	2	3	4	5	
5	3.4815	3.6523	3.8593	3.6896	3.6751	3.67156
10	3.3854	3.7794	3.776	3.8354	3.68	3.69124
20	3.5084	3.8358	3.7136	3.7188	3.6815	3.69162
25	3.7768	3.7427	3.6815	3.6754	3.673	3.70988
30	3.7919	3.7231	3.6895	3.7129	3.775	3.73848
50	3.4783	4.2023	3.5561	3.7269	3.7317	3.73906
60	3.9455	3.7884	3.7196	3.6389	3.6913	3.75674
70	3.9349	3.7819	3.7765	3.6688	3.6576	3.76394
80	3.9739	3.7824	3.6813	3.6843	3.7112	3.76662
90	3.9726	3.756	3.6954	3.6849	3.7401	3.7698
100	3.784	3.7856	3.7857	3.7829	3.7167	3.77098

Swarmsize	Waktu Komputasi Data Baik (s)
5	2.6
10	3.6
20	5.6
25	6
30	7
50	9.2
60	10
70	11.6
80	16.8
90	21.8
100	23.6

LAMPIRAN 9 HASIL PENGUJIAN MAKSIMAL ITERASI PSO

Hasil percobaan dengan data Premium

Maks. Iterasi	Nilai fitness pada percobaan ke-					Rata-rata fitness
	1	2	3	4	5	
10	4.1154	4.0756	3.9312	3.9265	3.7492	3.9596
20	4.1321	4.4295	4.6131	3.9041	4.0695	4.2297
30	3.3419	4.1102	4.4426	4.0540	6.3731	4.4644
40	4.1060	4.1270	6.3219	4.1142	4.7442	4.6827
50	4.0742	5.7752	3.3787	4.2396	6.0288	4.6993
70	5.9157	4.1423	4.6401	4.5250	4.3645	4.7175
90	5.2161	4.1382	4.8562	4.0052	5.6343	4.7700
100	4.2148	5.0419	6.3328	4.2479	4.1976	4.8070
120	5.3183	4.0501	4.3021	3.4921	6.259	4.6843
130	6.3443	4.6457	4.0405	4.1497	4.088	4.6536
150	4.0061	3.7671	4.0592	4.2697	4.0736	4.0351

Hasil percobaan dengan data Baik

Maks. Iterasi	Nilai fitness pada percobaan ke-					Rata-rata fitness
	1	2	3	4	5	
10	3.7562	3.7067	3.7290	3.8509	3.6815	3.74486
20	3.7471	3.7635	3.7741	3.6877	3.7814	3.75076
30	3.7799	3.7755	3.7832	3.7427	3.6815	3.75256
40	3.758	3.7806	3.7815	3.7219	3.7757	3.76354
50	3.7764	3.7307	3.7729	3.7711	3.7816	3.76654
70	3.776	3.7672	3.7415	3.7732	3.7762	3.76682
90	3.7725	3.7587	3.7361	3.8038	3.7674	3.76770
100	3.7776	3.7472	3.7716	3.7718	3.7724	3.76812
120	3.773	3.7766	3.7591	3.7717	3.7608	3.76824
130	3.7598	3.712	3.7493	3.7655	3.8548	3.76828
150	3.776	3.8744	3.7826	3.6257	3.7836	3.76846

LAMPIRAN 10 HASIL PENGUJIAN VARIASI OUTPUT

Hasil percobaan dengan data Premium

1. Kasus yang dioptimasi adalah protein optimal dan lemak minimal. Tabel hasil pengujian adalah sebagai berikut:

Tabel hasil pengujian Protein Optimal dan Lemak Minimal

Kasus	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
Protein Optimal dan Lemak Minimal	4.180	4.388	3.850	5.223	6.193	7.438	4.136	3.970	3.995	3.885	4.726

Tabel hasil kenaikan protein dan penurunan lemak

Kasus	Nilai fitness pada percobaan ke-										Rata-rata (%)
Protein Optimal dan Lemak Minimal	1	2	3	4	5	6	7	8	9	10	
Protein Naik	0.128	0.305	0.667	1.172	2.147	3.424	0.085	0.726	0.200	0.189	0.904
Lemak turun	0.339	1.759	2.997	0.271	0.815	5.193	0.269	2.217	0.168	1.241	1.527

2. Kasus variasi output yang digunakan adalah protein optimal dan laktosa optimal. Tabel hasil pengujian ditunjukkan sebagai berikut:

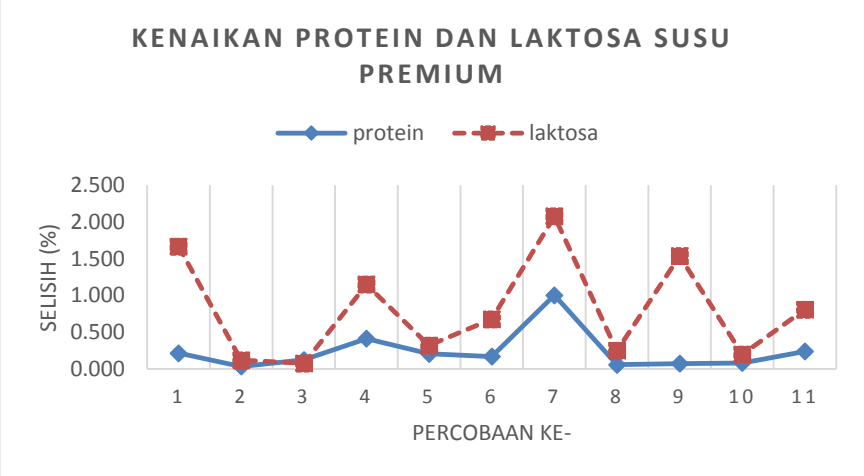
Tabel hasil pengujian protein dan laktosa optimal

Kasus	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
Protein Optimal dan Laktosa Optimal	7.054	8.779	9.129	7.371	8.818	8.424	12.007	9.121	10.541	9.046	9.029

Tabel hasil kenaikan protein dan laktosa

Kasus Protein Optimal dan Laktosa Optimal	Nilai fitness pada percobaan ke-										Rata-rata (%)
	1	2	3	4	5	6	7	8	9	10	
Kenaikan Protein	0.215	0.034	0.123	0.413	0.206	0.169	1.002	0.058	0.074	0.082	0.237
Kenaikan Laktosa	1.663	0.119	0.075	1.148	0.320	0.676	2.074	0.247	1.535	0.196	0.805

Grafik kenaikan protein dan laktosa pada susu premium



Hasil percobaan dengan data Baik

1. Kasus yang dioptimasi adalah protein optimal dan lemak minimal. Tabel hasil pengujian adalah sebagai berikut:

Tabel hasil pengujian Protein Optimal dan Lemak Minimal

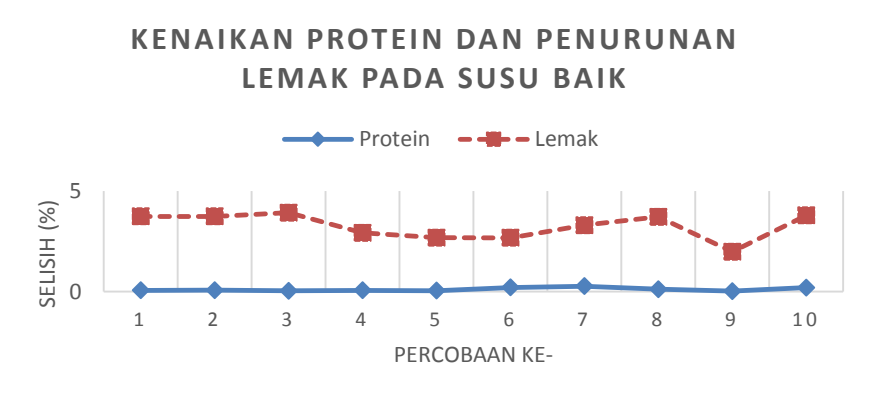
Kasus	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
Protein Optimal dan Lemak Minimal	3.720	3.836	3.696	3.792	3.808	3.693	3.682	3.879	3.733	3.674	3.720

Tabel hasil kenaikan protein dan penurunan lemak

Kasus	Nilai fitness pada percobaan ke-										Rata-rata (%)
Protein Optimal dan Lemak Minimal	1	2	3	4	5	6	7	8	9	10	
Protein Naik	4.18	4.39	3.85	5.22	6.19	7.44	4.14	3.97	4.00	3.88	4.18
Lemak turun	3.72	3.84	3.70	3.79	3.81	3.69	3.68	3.88	3.73	3.67	3.72

Grafik dari kenaikan protein dan penurunan lemak untuk kualitas susu Baik adalah sebagai berikut:

Grafik hasil pengujian kenaikan protein dan penurunan lemak



2. Kasus variasi output yang digunakan adalah protein optimal dan laktosa optimal. Tabel hasil pengujian ditunjukkan sebagai berikut:

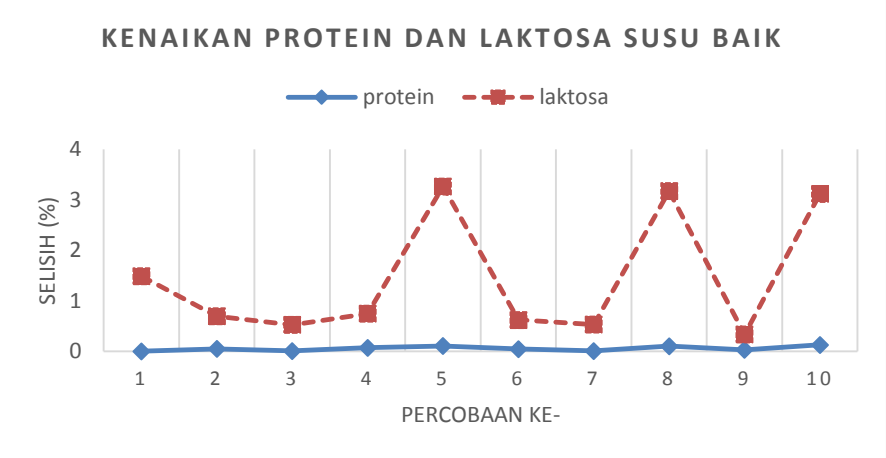
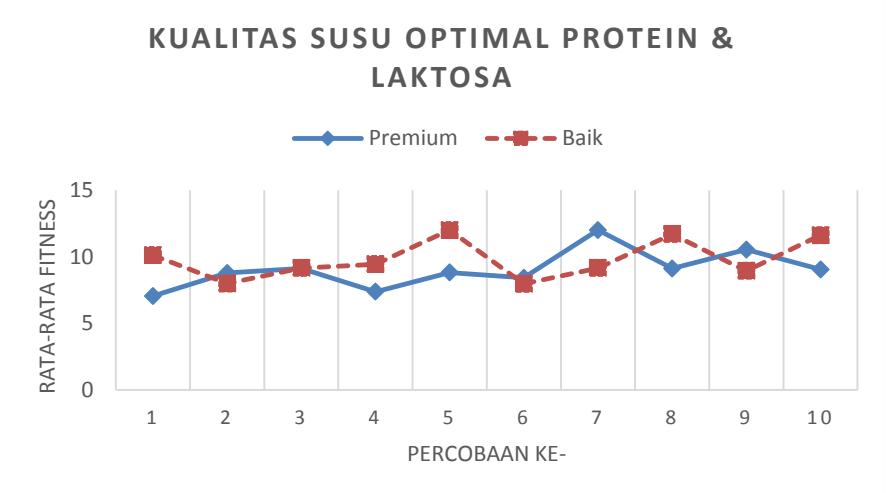
Tabel hasil pengujian protein dan laktosa optimal

Kasus	Nilai fitness pada percobaan ke-										Rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
Protein Optimal dan Laktosa Optimal	10.122	7.985	9.149	9.448	12.003	7.965	9.154	11.701	8.935	11.618	9.808

Tabel hasil kenaikan protein dan laktosa

Kasus Protein Optimal dan Laktosa Optimal	Nilai fitness pada percobaan ke-										Rata-rata (%)
	1	2	3	4	5	6	7	8	9	10	
Kenaikan Protein	0.003	0.049	0.008	0.071	0.107	0.045	0.008	0.102	0.031	0.129	0.055
Kenaikan Laktosa	1.487	0.696	0.526	0.746	3.264	0.621	0.530	3.171	0.334	3.116	1.449

Grafik kenaikan protein dan laktosa pada susu baik



Grafik hasil pengujian protein dan laktosa optimal