

## Optimasi Komposisi Pakan Kambing Boer Menggunakan Algoritme *Evolution Strategies*

Ramadhan Anindya Guna Aniwara<sup>1</sup>, Nurul Hidayat<sup>2</sup>, Tibyani<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>ramadhan.anindya11@gmail.com, <sup>2</sup>ntayadih@ub.ac.id, <sup>3</sup>tibyani@ub.ac.id

### Abstrak

Permasalahan yang sering terjadi dalam usaha ternak kambing boer adalah penggunaan bahan pakan yang belum efisien. Kesalahan dalam menentukan bahan pakan yang digunakan selama ini berdampak pada rendahnya kandungan nutrisi yang diberikan kepada kambing boer yang mengakibatkan kambing boer tersebut tidak tumbuh serta berkembang sama baik. Dalam sudut pandang ekonomi, pembelian bahan pakan ternak jadi biaya tertinggi dalam usaha peternakan, sehingga biaya tersebut harus ditekan serendah mungkin buat memaksimalkan penbisaan. Mengoptimalkan penyusunan bahan pakan (ransum) kambing boer adalah cara buat menekan biaya pembelian bahan pakan serta buat memaksimalkan keuntungan maupun penbisaan. Dalam kasus ini digunakan algoritme *evolution strategies* buat mengoptimasi komposisi bahan pakan kambing boer. Tipe proses ES yang digunakan adalah  $(\mu/r + \lambda)$  sehingga proses reproduksi melibatkan rekombinasi serta hasil mutasi nantinya akan diikutkan bersama *parent* dalam proses seleksi serta seleksi menggunakan *elitism selection*. Solusi optimal diperoleh dari ukuran populasi sebanyak 90 serta jumlah generasi 100 memperoleh rata-rata nilai *fitness* tertinggi yaitu 0.803. Hasil akhir berupa rekomendasi komposisi pakan ternak kambing boer.

**Kata kunci:** *Evolution Strategies, optimasi pakan, kambing boer.*

### Abstract

*The common problem that mostly happened in animal husbandry of goat is the insufficient usage of fodder. A simple mistake in choosing the fodder caused the goat to having a little nutrition in their body. That surely make the goat not growing breed well. In the economic point of view, the purchase of fodder becomes the highest cost in factory farming, so the cost should be reduced as low as possible to maximize the revenue. Optimizing the preparations of goat's fodder is a way to reducing the cost, that can also increasing the revenue. In this case, a evolution strategies algorithm is used to optimizing the goat's fodder. The type of ES process used is  $(\mu / r + \lambda)$  so that the reproduction process involves recombination and the mutation result will be included with the parent in the selection and selection process using elitism selection. The optimal solution is obtained from population size of 90 and the number of generations 100 obtained the highest average fitness value of 0.803. The final result is recommendation of goat's fodder.*

**Keywords:** *evolutions strategie, Optimizing the preparations of goat's fodder, feed of goat*

## 1. PENDAHULUAN

### 1.1 Latar belakang

Kambing boer adalah jenis kambing pedaging dengan pertumbuhan sangat pesat jika dibandingkan dengan jenis perah lokal dengan presentase daging lebih tinggi sebesar 40%-50% dari berat tubuhnya. Menurut Badan Pusat Statistik tahun 2017

konsumsi daging kambing mengalami peningkatan dari tahun 2015 pada angka 64.948 ekor pertahun menjadi 66.753 ekor pada tahun 2016.

Kesalahan pemberian pakan menjadi salah satu faktor yang menyebabkan kegagalan dalam usaha peternakan, karena selain pemberian bahan pakan adalah pengeluaran terbesar dalam usaha ternak nutrisi pakan juga menjadi penentu

keberhasilan usaha peternakan. Menekan pembelian bahan pakan dengan tetap menghitung kebutuhan nutrisi pakan kambing diperlukan untuk meningkatkan produksi serta keuntungan bagi peternak.

Dalam kasus ini digunakan metode *Evolution Strategies* untuk menyelesaikan permasalahan optimasi nutrisi bahan pakan kambing. *Evolution Strategies* dapat menyelesaikan permasalahan optimasi yang kompleks dengan parameter yang lebih sedikit. (Mahmudy, 2013).

Karena algoritme *evolution strategies* dapat menyelesaikan permasalahan optimasi maka pada skripsi ini algoritme *evolution strategies* akan diterapkan untuk membangun aplikasi “Optimasi Komposisi Pakan Kambing Boer Menggunakan Algoritme *Evolution Strategies*” dengan harapan aplikasi ini dapat berguna untuk para peternak agar dapat menekan biaya pembelian bahan pakan tetapi tetap memenuhi kebutuhan nutrisi kambing harian.

## 1.2 Rumusan Masalah

Berdasarkan sama latar belakang di atas maka disusun rumusan masalah sebagai berikut:

1. Bagaimana menerapkan *evolution strategies* pada aplikasi?
2. Bagaimana hasil pengujian dari algoritme *evolution strategies*?

## 1.3 Tujuan

Tujuan penelitian ini diantaranya:

1. Menerapkan algoritme *evolution strategies* pada aplikasi optimasi komposisi pakan kambing Boer.
2. Menguji hasil evaluasi penerapan algoritme *evolution strategies* pada sistem.

## 1.4 Batasan Masalah

Berdasarkan latar belakang maka disusun batasan masalah dalam pelaksanaan penelitian ini sebagai berikut:

1. Jenis kambing yang diteliti adalah jenis kambing Boer.

2. Jenis pakan yang digunakan dalam penelitian 25 jenis bahan pakan.

## 2. TINJAUAN PUSTAKA

### 2.1 Pemberian Pakan Kambing Boer

Kambing dan domba adalah hewan ternak dengan pakan utama berupa rumput hijau dan sejenisnya. Tetapi pada daerah tropis sering dijumpai rumput atau hijauan yang memiliki kualitas yang rendah, karena hal tersebut penambahan konsentrat dirasa butuh untuk meningkatkan nilai nutrisi dari bahan pakan. Pemberian serat pakan pada ternak dikatakan rumit dikarenakan kapasitas dari perut ternak yang biasanya membatasi asupan pakan sehingga kebutuhan nutrisinya tidak terpenuhi Ørskov (1998).

Kebutuhan pakan kambing bervariasi sesuai dengan fisiologis ternak Ludgate dkk (1989). Namun secara umum kebutuhan pakan kambing kurang lebih 10% dari berat badan ternak. Dengan contoh jika berat ternak 30 Kg maka diperlukan pakan sebesar 3 kg.

### 2.2 *Evolution Strategies*

*Evolution Strategies* pertama kali dikembangkan pada tahun 1960-an oleh mahasiswa *Technical University of Berlin (TUB)* yaitu Ingo Rechenberg serta Hans-Paul Schwefel. Konsep yang mendasari timbulnya ES adalah konsep evolusi serta seleksi alam yang dikemukakan oleh Charles Darwin. Fungsi dari ES adalah buat menemukan solusi optimal dari suatu permasalahan (Beyer, 2002).

*Evolution Strategies (ES)* memiliki siri utama berupa penggunaan vektor bilangan pecahan sebagai representasi solusi. Berbeda sama *Genetic Algorithms (GA)* yang menggunakan *crossover* sebagai operator reproduksi utama serta mutasi sebagai operator penunjang. ES lebih bertumpu pada operator mutasi. Mekanisme *self-adaptation* digunakan buat mengontrol perubahan nilai parameter pencarian. GA serta ES bisa digunakan buat menyelesaikan permasalahan yang sama. Tetapi mana yang terbaik di antara kedua metode tersebut sangat tergantung pada permasalahan yang dihadapi (Mahmudy, 2013).

### 2.3 Representasi kromosom

Representasi kromosom adalah proses

pengkodean dari penyelesaian asli suatu permasalahan. Solusi dari suatu permasalahan harus dipetakan (*encoding*) jadi string kromosom. String kromosom tersusun atas sejumlah gen yang menggambarkan variable-variabel keputusan yang digunakan dalam solusi (Mahmudy, 2013).

Seperti halnya buat *real-coded* GA, variable keputusan (x1 serta x2) langsung jadi gen string *chromosome*. Selain gen yang menyatakan variable keputusan, parameter tambahan yang melekat pada setiap *chromosome* adalah *sigma* ( $\sigma$ ). Nilai ini menyatakan level mutasi buat *chromosome* tersebut. Nilai ini akan berubah sepanjang generasi secara adaptif. Jika P adalah satu chromosome maka  $P = (x1, x2, \sigma1, \sigma2)$  sama panjang string sebesar 4.

**2.4 Inisialisasi**

Tahap inisialisasi ini bertujuan buat membangkitkan himpunan solusi baru secara acak yang terdiri dari sejumlah string kromosom serta ditempatkan pada penampungan yang disebut sama populasi. Pada tahap ini *miu* ( $\mu$ ) yang menyatakan ukuran populasi harus ditentukan. Panjang string kromosom (*stringLen*) dihitung berdasarkan presisi variable serta solusi yang dicari (Mahmudy, 2013).

**2.5 Reproduksi**

Reproduksi bertujuan buat menghasilkan keturunan dari individu-individu yang ada di populasi. Berbeda pada algoritme genetik yang menggunakan dua operator genetika yaitu tukar silang (*crossover*) serta mutasi (*mutation*) pada algoritme *evolution strategies* menggunakan rekombinasi serta mutasi.

**2.5.1 Rekombinasi**

Rekombinasi dilakukan buat menghasilkan *offspring* sebanyak dari sejumlah  $\mu$  individu dalam populasi. Setiap satu individu *offspring* dihasilkan dari beberapa induk. Induk dipilih secara acak dari populasi. Metode rekombinasi paling sederhana adalah sama menghitung nilai rata-rata nilai elemen induk. Contoh proses rekombinasi diberikan sebagai berikut :

- Misalkan *offspring* diprolehkan dari 2 induk. Jika P1 serta P3 terpilih maka akan diprolehkan *offspring*  $C = (-0,17815, 1,90205, 0,57016, 0,70221)$ .

- Misalkan *offspring* diprolehkan dari 3 induk. Jika P1, P2 serta P3 terpilih maka akan diprolehkan *offspring*  $C = (2,71180, 2,12650, 0,55944, 0,75782)$

Pada studi kasus ini, misalkan  $\lambda = 6$  serta *offspring* diprolehkan dari 2 induk.

**2.5.2 Mutasi**

Mutasi biasanya digunakan sebagai operator buat menjaga keragaman populasi. Misalkan  $P = (x1, x2, \sigma1, \sigma2)$  adalah individu yang terpilih buat melakukan mutasi, maka dihasilkan *offspring*  $P' = (x1', x2', \sigma1', \sigma2')$  sebagai berikut :

$$x' = x + \sigma N(0,1)$$

Rumusan ini bisa didetailkan sebagai berikut:

$$x' = x1 + \sigma1 N(0,1)$$

$$x' = x2 + \sigma2 N(0,1) \dots\dots\dots (2-1)$$

$N(0,1)$  adalah bilangan acak yang mengikuti sebaran normal sama rata-rata sebesar 0 serta standard deviasi sebesar 1. Pada program komputer, nilai  $N(0,1)$  bisa diprolehkan sama membangkitkan dua bilangan r1 serta r2 pada interval [0,1]. Rumus yang digunakan adalah (Schwefel, 1995):

$$N(0,1) = \sqrt{-2 \ln r1 \sin 2\pi r2} \dots\dots\dots (2-2)$$

Misalkan r1 = 0,4749 serta r2 = 0.3296 maka diprolehkan  $N(0,1) = 1,0709$

Nilai  $\sigma$  dinaikkan jika *fitness* hasil mutasi lebih baik dari induknya. Sebaliknya jika *fitness* hasil mutasi lebih buruk dari induknya maka nilai  $\sigma$  diturunkan. Nilai  $\sigma$  dinaikkan sama rumusan  $\sigma' = \sigma \times 1,1$  serta diturunkan sama rumusan  $\sigma' = \sigma \times 0,9$  (Mahmudy, 2013)

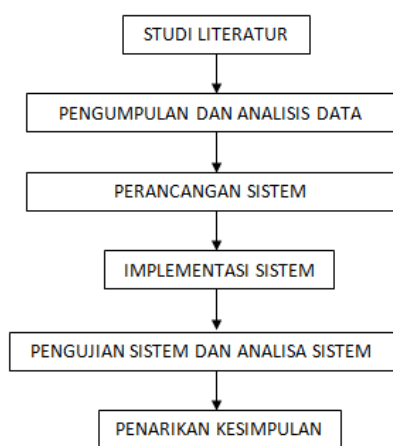
**2.6 Seleksi**

Seleksi adalah tahapan terakhir yang dilakukan buat memilih individu dari himpunan populasi serta *offspring* yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar nilai *fitness* kromosom, maka semakin besar peluang kromosom tersebut terpilih. Metode seleksi *elitism* bekerja sama cara mengumpulkan semua individu dalam populasi (*parent*) serta *offspring* dalam satu penampungan. Metode ini melakukan seleksi pada individu-individu dalam penampungan berdasarkan nilai *fitness* tertinggi. Individu terbaik dalam penampungan akan lolos buat masuk dalam generasi berikutnya. Metode seleksi *elitism* menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013).

### 3. METODOLOGI PENELITIAN

Pada bab metode penelitian serta perancangan ini akan dibahas langkah-langkah yang digunakan dalam pembuatan aplikasi optimasi komposisi pakan kambing potong/pedaging sama menggunakan algoritme genetika. Adapun langkah-langkahnya adalah sebagai berikut :

1. Melakukan studi literatur tentang algoritma genetika serta penyusunan atau formulasi pakan kambing potong/pedaging yang akan dibuat serta dibahas pada skripsi ini.
2. Menganalisa serta merancang system sama menggunakan hasil dari studi literature tersebut.
3. Membuat sistem berdasarkan analisa serta perancangan yang telah dilakukan.
4. Membuat uji coba terhadap sistem yang telah dibuat.
5. Melakukan evaluasi hasil terhadap hasil yang diperoleh dari uji coba tersebut.



Gambar 3.1 Tahapan Penelitian

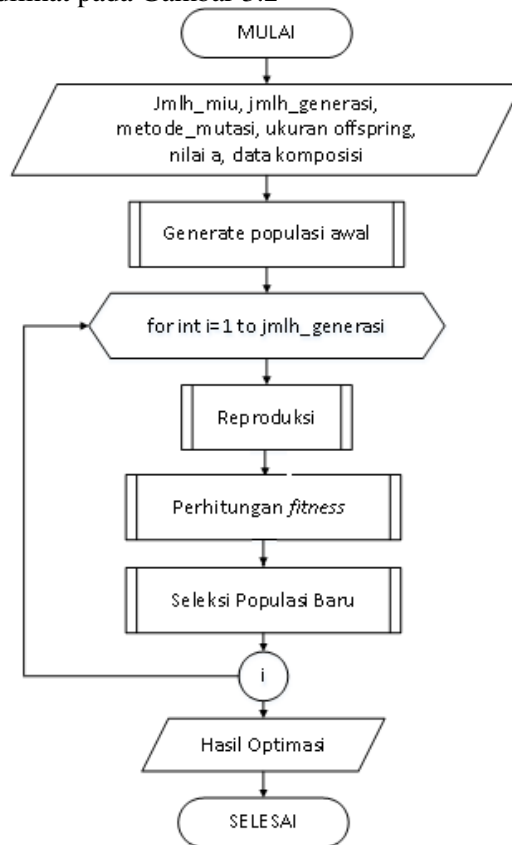
#### 3.1 Data Penelitian

Data yang digunakan dalam penelitian adalah sebagai berikut:

1. Daftar kebutuhan nutrisi kambing boer dari penelitian mahasiswa jurusan peternakan universitas brawijaya.
2. Daftar harga bahan makanan diprolehkan berdasarkan survei di Kota Malang pada tahun 2014

### 3.2 Perancangan Sistem

Siklus penyelesaian pembentukan susunan ransum optimal menggunakan algoritme ES bisa dilihat pada Gambar 3.2



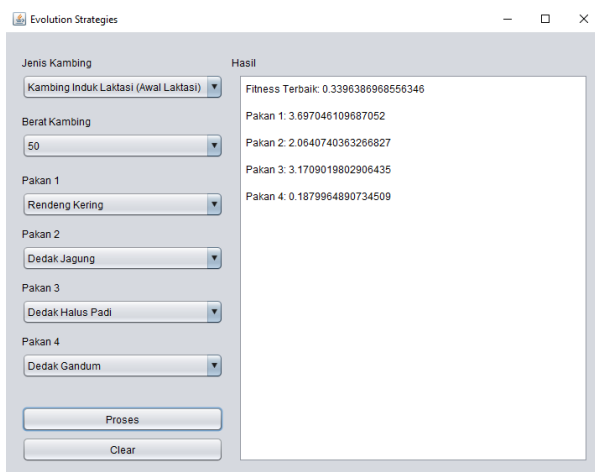
Gambar 3.2 Diagram Alir Evolution Strategies

Seperti yang terlihat pada Gambar 3.2 terbiasa tiga proses utama buat menyelesaikan masalah optimasi komposisi pakan ternak sapi potong, yaitu proses *Generate Populasi Awal*, *Reproduksi*, *Perhitungan Fitness* serta *Seleksi Populasi Baru*. Langkah kerjanya adalah setelah memilih kategori sapi beserta bobot baserta serta penambahan bobot baserta harian yang diinginkan, menginputkan bahan pakan serta parameter ES, maka sistem akan membuat populasi awal sesuai sama jumlah yang diinputkan. Setelah populasi awal terbentuk akan dilakukan pengulangan buat membentuk *offspring* sama cara rekombinasi serta mutasi. Hasil mutasi serta populasi awal akan dihitung nilai *fitness* masing-masing individu lalu diseleksi. Proses pengulangan dilakukan sampai banyak generasi tercapai. Hasil keluaran berupa hasil optimasi yang diprolehkan sama algoritme *evolution strategies*.

### 4. IMPLEMENTASI

### 4.1 Implementasi Antarmuka

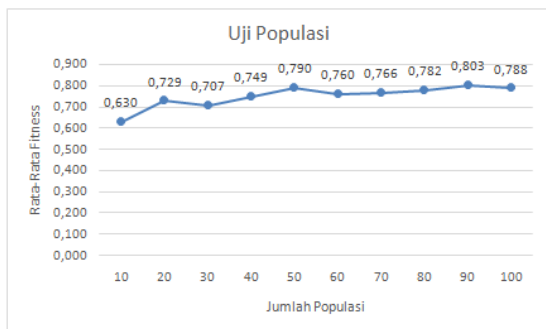
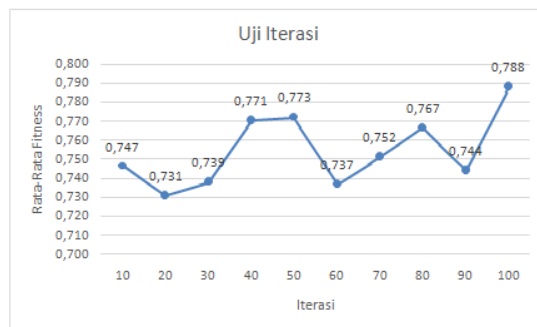
Implementasi Antarmuka pada aplikasi ini hanya berupa satu halaman tampilan utama yang berisi enam buah *combobox* yang berisi jenis kambing, berat dari kambing, dan empat buah *combobox* yang masing-masing berisi jenis dari bahan pakan. Kemudian ada satu *textfield* kosong yang nantinya akan memunculkan hasil akhir dari perhitungan optimasi. Tampilan antarmuka ditunjukkan pada gambar 4.1.



Gambar 4.1 Implementasi Antarmuka Halaman Utama

Pengujian pertama dilakukan terhadap ukuran populasi. Pengujian ini ditujukan untuk mengetahui pengaruh ukuran populasi pada hasil *fitness* dari perhitungan dalam aplikasi ini. Pengujian dilakukan dengan cara melakukan perhitungan sebanyak 10 kali kemudian diambil nilai dari rata-ratanya. Ukuran populasi yang diuji mulai dari 10 sampai dengan 100 dengan *range* berkelipatan 10. Hasil pengujian ukuran populasi dapat dilihat pada gambar 5.1.

Dapat dilihat pada gambar 5.1 pada pengujian ukuran populasi didapatkan hasil yang kurang stabil tetapi masih memiliki kecenderungan peningkatan hasil *fitness* sebanding dengan bertambahnya ukuran populasi, terbukti hasil *fitness* terbaik didapat pada ukuran populasi 90 dengan nilai *fitness* 0.803.



Gambar 5.1 Hasil Uji Ukuran Populasi

### 5.3 Pengujian dan Analisis Jumlah Iterasi

Pengujian parameter jumlah iterasi kedua dilakukan untuk melihat apakah jumlah iterasi mempengaruhi hasil *fitness* dari perhitungan. Pengujian dilakukan dengan cara mencari nilai rata-rata dari pengujian sebanyak 10 kali masing-masing jumlah iterasi pada sistem dengan ketentuan jumlah iterasi berkelipatan 10 mulai dari 10 hingga 100. Hasil dari pengujian dapat dilihat pada gambar 5.2.

Dapat dilihat pada gambar 5.2 pada pengujian jumlah iterasi didapatkan hasil yang kurang stabil tetapi masih memiliki kecenderungan peningkatan hasil *fitness* sebanding dengan bertambahnya jumlah iterasi, terbukti hasil *fitness* terbaik didapat pada ukuran populasi 100 dengan nilai *fitness* 0.788.

### 5.4 Pengujian dan Analisis Jumlah Lamda

## 5. PENGUJIAN DAN ANALISA

### 5.1 Analisis dan Pembahasan

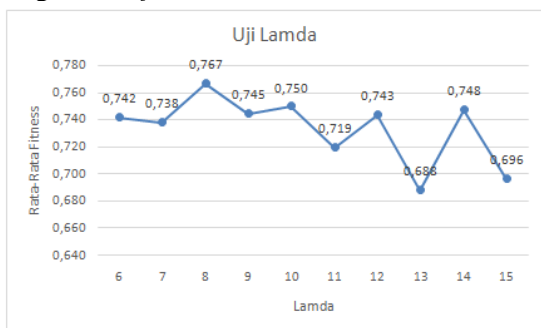
Pengujian pada penelitian ini meliputi 3 tahap yaitu pengujian ukuran populasi, pengujian jumlah iterasi, dan pengujian jumlah lamda.

### 5.2 Pengujian dan Analisis Ukuran Populasi



Pengujian parameter jumlah lamda dilakukan untuk melihat apakah jumlah lamda mempengaruhi hasil *fitness* dari perhitungan. Pengujian dilakukan dengan cara mencari nilai rata-rata dari pengujian sebanyak 10 kali masing-masing jumlah lamda pada sistem dengan ketentuan jumlah iterasi berkelipatan 1 mulai dari 6 hingga 15. Hasil dari pengujian dapat dilihat pada gambar 5.3.

Dapat dilihat pada gambar 5.3 pada pengujian jumlah lamda didapatkan hasil yang kurang stabil dan memiliki kecenderungan penurunan hasil *fitness* sebanding dengan bertambahnya jumlah lamda, terbukti hasil *fitness* terbaik didapat pada ukuran populasi 3 dengan nilai *fitness* 0.767.



Gambar 5.3 Grafik Pengujian Jumlah Lamda

6. KESIMPULAN

Kesimpulan yang diperoleh melalui hasil uji coba yang telah dilakukan mengenai penerapan algoritme evolution strategies buat menyelesaikan permasalahan optimasi komposisi pakan buat kambing yaitu sebagai berikut:

1. Algoritme evolution strategies bisa diterapkan pada permasalahan optimasi pakan buat kambing sama menggunakan menggunakan *exchange mutation* serta penyeleksian sama metode *elitismselection*.
2. Pengujian parameter algoritme evolution strategies memiliki pengaruh terhadap nilai *fitness* yang dihasilkan. Pengujian pada parameter evolution strategies dilakukan sama melakukan perubahan jumlah populasi dari 10 hingga 100 serta menghasilkan fitness sebesar 0,803 pada jumlah populasi 90. Parameter algoritme evolution strategies yang juga diuji terhadap penelitian ini yaitu generasi serta populasi. Kedua

parameter tersebut berpengaruh terhadap nilai *fitness* yang diperoleh dari proses algoritme evolution strategies. Ukuran populasi yang optimal terbiasa pada ukuran 90 sama nilai *fitness* sebesar 0.803.

7. DAFTAR PUSTAKA

Ludgate, P.J.,dkk.1989. *Kumpulan Peragaan Dalam Rangka Penelitian Ternak Kambing serta Domba di Pedesaan*. Balai Penelitian Ternak. Pusat Penelitian serta Pengembangan Peternakan. Baserta Litbang Pertanian. Bogor.

Ørskov, E.R. 1998. *The Feeding of Ruminants Principles And Practice*. Aberdeen: Rowett Research Institute.

Pramesti, D, Mahmudy, WF & Indriati 2015.*Optimasi komposisi pakan kambing boer menggunakan algoritme genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, vol. 5, no. 13.

Mahmudy, WF. 2013. *Algoritme Evolusi*. Program Teknologi Informasi serta Ilmu Komputer. Universitas Brawijaya. Malang