

BAB 4 PERANCANGAN

Pada bab perancangan ini membahas tentang langkah penyelesaian masalah yang dibahas menggunakan metode yang diajukan, yaitu *evolution strategies*. Langkah penyelesaian masalah menggunakan *evolution strategies*, perancangan antarmuka serta perancangan uji coba yang akan digunakan dalam pembuatan aplikasi optimasi komposisi pakan kambing boer menggunakan *evolution strategies*.

4.1 Struktur dasar Evolution Strategies (ES)

Prosedur umum dalam ES dapat dinyatakan dengan istilah (μ, λ) . Di mana μ adalah jumlah solusi awal atau populasi awal, sedangkan λ merupakan jumlah solusi yang dihasilkan dari generasi awal (*offspring*). Karena ES lebih mengandalkan mutasi, maka proses rekombinasi tidak selalu digunakan. Secara umum terdapat empat tipe proses dari ES, yaitu :

- (μ, λ)
- $(\mu/r, \lambda)$
- $(\mu + \lambda)$
- $(\mu/r + \lambda)$

$ES(\mu, \lambda)$ tidak menggunakan rekombinasi dalam proses reproduksi. Seleksi menggunakan *elitism selection* hanya melibatkan individu dalam offspring, individu induk dalam populasi tidak dilibatkan. $ES(\mu/r, \lambda)$ serupa dengan $ES(\mu, \lambda)$ dengan tambahan melibatkan proses rekombinasi. $ES(\mu + \lambda)$ tidak menggunakan rekombinasi dan proses seleksi menggunakan *elitism selection* melibatkan individu offspring dan induk (Mahmudy, 2013).

4.1.1 Reprerentasi kromosom

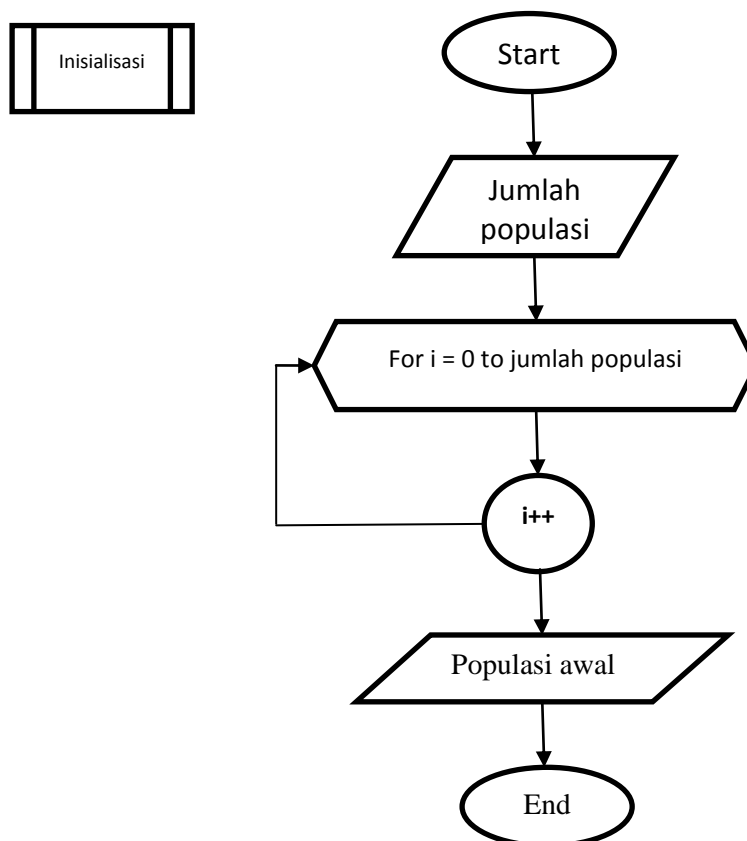
Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Solusi dari suatu permasalahan harus dipetakan (*encoding*) menjadi string kromosom. String kromosom tersusun atas sejumlah gen yang menggambarkan variable-variabel keputusan yang digunakan dalam solusi (Mahmudy,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

Seperti halnya untuk *real-coded* GA, variable keputusan (x_1 dan x_2) langsung menjadi gen string *chromosome*. Selain gen yang menyatakan variable keputusan, parameter tambahan yang melekat pada setiap *chromosome* adalah *sigma* (σ). Nilai ini menyatakan level mutasi untuk *chromosome* tersebut. Nilai ini akan berubah sepanjang generasi secara adaptif. Jika P adalah satu chromosome maka $P = (x_1, x_2, \sigma_1, \sigma_2)$ dengan panjang string sebesar 4.

4.1.2 Inisialisasi

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

Proses Inisialisasi dapat dilihat pada gambar 4.1.



4.1.3 Reproduksi

Reproduksi bertujuan untuk menghasilkan keturunan dari individu-individu yang ada di populasi. Berbeda pada algoritme genetik yang menggunakan

dua operator genetika yaitu tukar silang (*crossover*) dan mutasi (*mutation*) pada algoritme *evolution strategies* menggunakan rekombinasi dan mutasi.

4.1.3.1 Rekombinasi

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

- Misalkan *offspring* didapatkan dari 2 induk. Jika P1 dan P3 terpilih maka akan didapatkan *offspring* C=(-0,17815, 1,90205, 0,57016, 0,70221).
- Misalkan *offspring* didapatkan dari 3 induk. Jika P1, P2 dan P3 terpilih maka akan didapatkan *offspring* C = (2,71180, 2,12650, 0,55944, 0,75782)

4.1.4 Mutasi

Mutasi biasanya digunakan sebagai operator untuk menjaga keragaman populasi. Misalkan $P = (x_1, x_2, \sigma_1, \sigma_2)$ adalah individu yang terpilih untuk melakukan mutasi, maka dihasilkan *offspring* $P' = (x_1', x_2', \sigma_1', \sigma_2')$ sebagai berikut :

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

Rumusan ini bisa didetailkan sebagai berikut:

$$\begin{aligned} x' &= x_1 + \sigma_1 N(0,1) \\ x' &= x_2 + \sigma_2 N(0,1) \dots\dots\dots (2-1) \end{aligned}$$

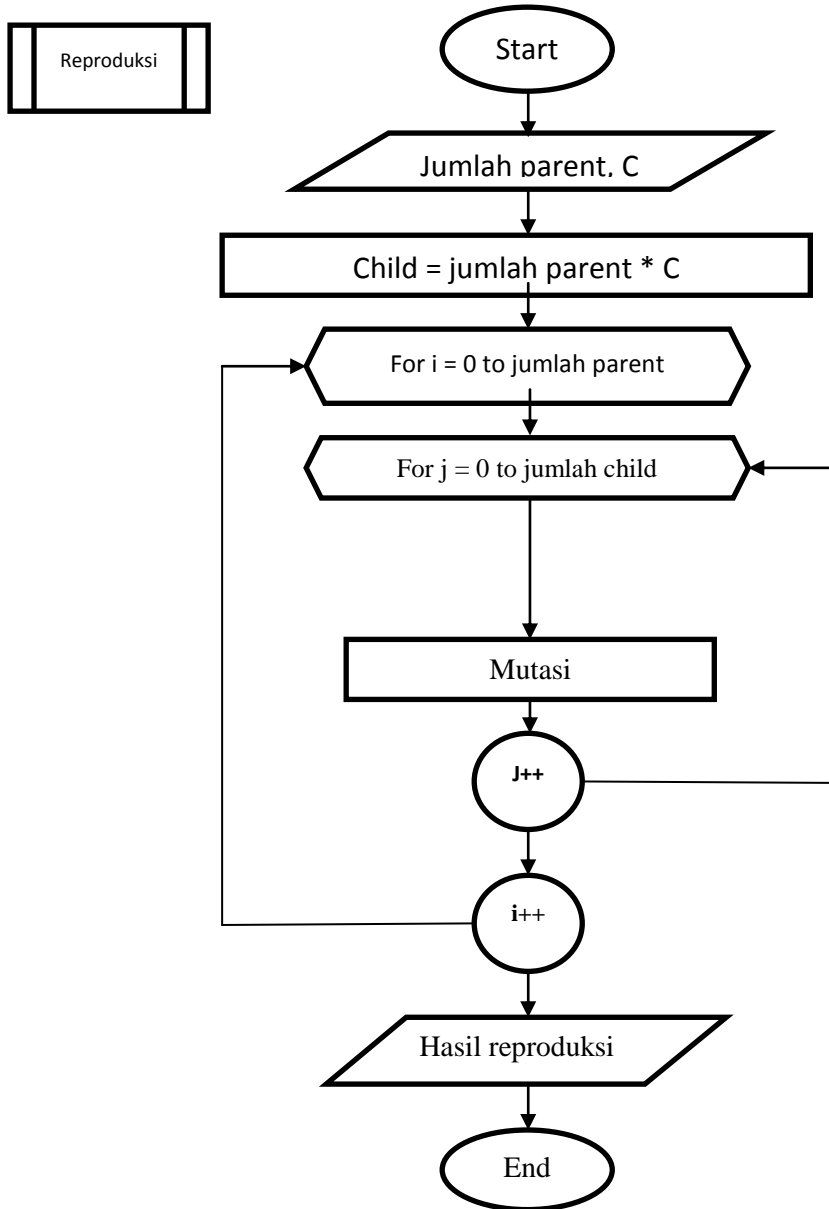
$N(0,1)$ merupakan bilangan acak yang mengikuti sebaran normal dengan rata-rata sebesar 0 dan standard deviasi sebesar 1. Pada program komputer, nilai $N(0,1)$ bisa didapatkan dengan membangkitkan dua bilangan r_1 dan r_2 pada interval $[0,1]$. Rumus yang digunakan adalah (Schwefel, 1995):

$$N(0,1) = \sqrt{-2 \ln r_1 \sin 2\pi r_2} \dots\dots\dots (2-2)$$

Misalkan $r_1 = 0,4749$ dan $r_2 = 0,3296$ maka didapatkan $N(0,1) = 1,0709$

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

Proses reproduksi dapat dilihat pada gambar 4.2.



Gambar 4.2 Diagram alir Reproduksi

4.1.5 Penghitungan Fitness

Algoritme genetika adalah algoritma yang dapat menyelesaikan masalah kompleks, salah satunya yaitu permasalahan optimasi.

Dalam permasalahan optimasi harus didapatkan nilai *fitness*, yaitu nilai dari suatu individu yang menjadi tolak ukur dalam pengevaluasian berdasarkan suatu nilai tertentu. Apabila permasalahannya adalah optimasi untuk memaksimalkan fungsi h , maka akan memakai fungsi $f = h$ dalam mencari nilai *fitnessnya*. Tapi jika masalah yang ingin diselesaikan berupa optimasi untuk meminimalkan fungsi h , maka akan memakai fungsi $f = 1/h$ dalam mendapatkan nilai *fitnessnya*. Jika h bernilai 0 maka akan membuat nilai f bernilai tak terhingga. Sehingga nilai h pada fungsi $f = 1/h$ harus ditambah dengan bilangan yang dianggap kecil. Sehingga rumus *fitness* ditunjukkan pada Persamaan (2-3) (Kushardiana, 2013).

$$f = \frac{1}{(h+a)} \quad (2-3)$$

Dimana :

- α adalah bilangan kecil atau konstanta yang sesuai dengan permasalahan yang dihadapi.
- h adalah sebuah fungsi yang akan diminimalkan (dalam kasus ini adalah harga bahan pakan).
- f adalah fungsi *fitness*

Pada penelitian *fitness* yang dicari adalah komposisi pakan dengan nutrisi terbaik tapi memiliki harga termurah, oleh karena itu variabel yang dibutuhkan adalah total nutrisi, penalty dan total harga. Total nutrisi dapat dihitung dengan persamaan (2-4).

$$\text{Nutrisi} = \text{berat bahan} \times \frac{\text{Nilai nutrisi}}{100} \quad (2-4)$$

Nilai penalty didapatkan dari perbandingan antara nutrisi pada ransum dan nutrisi yang dibutuhkan, jika nutrisi ransum lebih dari nutrisi minimal yang dibutuhkan maka nilai penalty dianggap nol tetapi jika nutrisi dari ransum kurang dari nutrisi minimal yang dibutuhkan maka akan masuk ke persamaan (2-5).

$$\text{Penalty} = -(\text{nutrisi ransum} - \text{nutrisi minimal}) \quad (2-5)$$

Sedangkan untuk penghitungan harga dapat dihitung menggunakan persamaan (2-6).

$$\text{Harga} = \text{Berat bahan} \times \text{harga perkilo} \quad (2-6)$$

Setelah didapatkan nilai nutrisi dan harga untuk setiap gen kemudian dapat menghitung nilai *fitness* menggunakan persamaan (2-7).

$$\text{Fitness} = \frac{10000}{\text{total harga} + (\text{penalty} \times 100000)} \quad (2-7)$$

4.1.6 Seleksi

Seleksi merupakan tahapan terakhir yang dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang akan dipertahankan hidup pada generasi berikutnya. Semakin besar nilai *fitness* kromosom, maka semakin besar

peluang kromosom tersebut terpilih. Metode seleksi *elitism* bekerja dengan cara mengumpulkan semua individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan. Metode ini melakukan seleksi pada individu-individu dalam penampungan berdasarkan nilai *fitness* tertinggi. Individu terbaik dalam penampungan akan lolos untuk masuk dalam generasi berikutnya. Metode seleksi *elitism* menjamin individu yang terbaik akan selalu lolos (Mahmudy, 2013).

4.2 Contoh perhitungan manual

Terdapat bahan – bahan yang dapat dikomposisikan menjadi pakan, dengan nutrisi yang berbeda dan harga bahan yang berbeda- beda pula tentunya akan menghasilkan kombinasi yang beragam, seorang formulator dapat melihat komposisi nutrisi bahan pangan dan nutrisi yang dibutuhkan untuk setiap jenis kambing pada tabel 4.1 dan 4.2, diharapkan dengan *evolution strategies* ini akan menghasil komposisi ransum yang memiliki nutrisi terbaik dengan harga termurah.

Misal terdapat kambing pejantan dengan berat 30 kg. Maka kebutuhan nutrisinya adalah sebagai berikut :

$$\begin{aligned} - \text{Berat pakan perhari} &= \frac{10}{100} \times 30 \\ &= 3 \text{ kg} \end{aligned}$$

$$\begin{aligned} - \text{Ca minimal} &= \frac{0,2}{100} \times 30 \\ &= 0,06 \end{aligned}$$

$$\begin{aligned} - \text{P minimal} &= \frac{0,18}{100} \times 30 \\ &= 0,054 \end{aligned}$$

Bahan pakan yang tersedia diantaranya Gaplek, onggok, cantel dan tetesdimana berdasarkan tabel 4.2 bahan – bahan tersebut memiliki nutrisi sebagai berikut :

Gaplek : Ca 0,1% dan P 0,04%

Onggok : Ca 0,68% dan P 0,05%

Cantel : Ca 0,19% dan P 0,2%

Tetes : Ca 0,71% dan P 0,07%

4.2.1 Inisialisasi

Tahap ini adalah tahap awal untuk membangkitkan individu atau kromosom awal, pada tahap ini akan dibangkitkan kromosom sesuai jumlah inputan, pada kasus ini sebagai contoh akan dibangkitkan 5 populasi dan karena

masukannya sebanyak 4 macam bahan pakan maka setiap kromosom memiliki 4 gen yang merupakan masing-masing bahan yang akan dioptimasi.

Inisialisasi awal dapat dilihat pada tabel 4.3.

Tabel 4.1 Inisialisasi populasi

Populasi	Gaplek	Onggok	Cantel	Tetes
P1	2,323	2,564	2,885	1,773
P2	2,743	2,656	2,135	2,455
P3	2,254	2,887	2,223	1,767
P4	1,970	2,547	1,984	2,453
P5	1,906	2,644	1,854	1,556

4.2.2 Hitung fitness

Setelah tahap inisialisasi selanjutnya dilakukan penghitungan fitness untuk setiap kromosom, penghitungan fitness itu sendiri melalui beberapa tahap yaitu penghitungan harga, penghitungan nutrisi, penghitungan penalty dan terakhir penghitungan fitness.

4.2.2.1 Hitung harga

Salah satu tahap penghitungan fitness adalah menghitung total harga untuk setiap kromosom, penghitungan harga ini menggunakan persamaan 2-6. Hasil penghitungan harga ditunjukkan oleh tabel 4.4.

Tabel 4.2 inisialisasi populasi

Populasi	Gaplek	Onggok	Cantel	Tetes
P1	2,323	2,564	2,885	1,773
P2	2,743	2,656	2,135	2,455
P3	2,254	2,887	2,223	1,767
P4	1,97	2,547	1,984	2,453
P5	1,906	2,644	1,854	1,556

Tabel 4.3 harga dan total harga masing-masing kromosom

Populasi	Gaplek	Onggok	Cantel	Tetes	total
P1	5807,5	5128	20195	4432,5	35563
P2	6857,5	5312	14945	6137,5	33252
P3	5635	5774	15561	4417,5	31387,5
P4	4925	5094	13888	6132,5	30039,5
P5	4765	5288	12978	3890	26921

4.2.2.2 Hitung nutrisi

Tahap selanjutnya ialah proses penghitungan kandungan nutrisi pada setiap kromosom, penghitungan nutrisi ini menggunakan persamaan 2-4. Hasil dari perhitungan nutrisi setiap kromosom ditunjukkan oleh tabel 4.6.

Tabel 4.4 Hasil perhitungan nutrisi

Populasi	BK(%)	PK(%)	TDN(%)	Ca(%)	P(%)
P1	8,2087	0,493485	6,51716	0,037828	0,009222
P2	8,59054	0,447293	6,62822	0,042291	0,008414
P3	7,85266	0,425022	6,14672	0,038655	0,008028
P4	7,70044	0,414758	5,90214	0,040476	0,007747
P5	6,8456	0,36357	5,34162	0,034455	0,006882

4.2.2.3 Hitung penalty

Setelah didapatkan kandungan nutrisi setiap kromosom selanjutnya dilakukan penghitungan penalty, penghitungan penalty ini menyesuaikan dengan nutrisi yang dibutuhkan oleh kambing, untuk perbandingannya menggunakan persamaan 2-5. Jika nutrisi yang terdapat pada ransum melebihi nutrisi yang dibutuhkan oleh kambing maka penalty dianggap nol, tetapi jika nutrisi ransum kurang dari nutrisi yang dibutuhkan kambing maka akan digunakan persamaan 2-5 untuk menghitung penalty. Hasil penghitungan penalty ditunjukkan oleh tabel 4.7.

Tabel 4.5 hasil perhitungan penalty

Populasi	BK(%)	PK(%)	TDN(%)	Ca(%)	P(%)	total
P1	0	0	0	0	0	0
P2	0	0	0	0	0	0
P3	0	0	0	0	0	0
P4	0	0	0	0	0	0
P5	0	0	0	0	0	0

Karena keseluruhan nutrisi ransum melebihi nutrisi yang dibutuhkan kambing maka nilai penalty keseluruhan bernilai 0.

4.2.2.4 Hitung fitness

Setelah didapatkan total harga masing-masing kromosom beserta total penalty kemudian masuk ke proses perhitungan fitness dengan persamaan 2-7.

Hasil perhitungan fitness ditunjukkan oleh tabel 4.8

Tabel 4.6 Hasil perhitungan fitness

Populasi	Gaplek	Onggok	Cantel	Tetes	harga	penalty	fitness
P1	2,323	2,564	2,885	1,773	35563	0	0,28119113
P2	2,743	2,656	2,135	2,455	33252	0	0,30073379
P3	2,254	2,887	2,223	1,767	31387,5	0	0,31859817
P4	1,97	2,547	1,984	2,453	30039,5	0	0,33289502
P5	1,906	2,644	1,854	1,556	26921	0	0,37145723

4.2.3 Reproduksi

Pada penelitian ini untuk reproduksi hanya menggunakan mutasi tanpa adanya rekombinasi.

4.2.3.1 Mutasi

Setelah dilakukan perhitungan fitness masuk ke tahap mutasi, evolution strategies sangat berfokus pada proses mutasi dimana proses mutasi ini adalah salah satu proses utama untuk mendapatkan individu baru dengan fitness melebihi induk atau parent. Contoh perhitungan manual ini λ bernilai 2 sehingga masing-masing parent menghasilkan 2 child.

Proses pertama adalah untuk mendapatkan c1, individu c1 didapatkan dari dua gen parent(pada kasus ini parentnya adalah p1) yang dipilih secara acak kemudian ditukar. Gen akan dipilih dan diacak sebanyak $1 * \vartheta$ kali, nilai ϑ mula-mula adalah random dari 1-3, jika individu baru yang dihasilkan memiliki nilai fitness yang lebih tinggi dari fitness induk maka untuk inisialisasi selanjutnya nilai ϑ bernilai tinggi yaitu 1,1 tetapi jika individu baru yang dihasilkan memiliki fitness yang lebih buruk maka untuk inisialisasi selanjutnya ϑ bernilai rendah yaitu 0,9. Nilai ϑ akan selalu dibulatkan keatas (misal ϑ bernilai 0,2 maka nilai ϑ dianggap 1). Individu yang dihasilkan ditunjukkan oleh tabel 4.14.

Tabel 4.7 inisialisasi kromosom c3

populasi	asal	gaplek	onggok	cantel	tetes	ϑ
c1	p1	2,5640	2,3230	1,7730	2,8850	2

Setelah didapatkan inisialisasi kemudian langsung dihitung fitness untuk c1. Hasil dari perhitungan fitness ditunjukkan oleh tabel 4.15.

Tabel 4.8 Fitness c3

populasi	asal	gaplek	onggok	cantel	tetes	ϑ	fitness
c1	p1	2,5640	2,3230	1,7730	2,8850	2	0,3260

Setelah didapatkan fitness c1 maka kemudian fitness c1 dibandingkan dengan fitness induk yaitu p1, karena fitness c1 lebih dari fitness p1 maka untuk individu selanjutnya dengan parent p1 ϑ akan bernilai $\vartheta * 1,1$.

Selanjutnya adalah inisialisasi untuk c2, hasil dari inisialisasi c2 dapat dilihat pada tabel 4.16.

Tabel 4.9 inisialisasi c4

populasi	asal	gapplek	onggok	cantel	tetes	ϑ
c2	p1	1,7730	2,8850	2,5640	2,3230	2,2

Gen c2 didapat dari dari dua gen parent(pada kasus ini parentnya adalah p1) yang dipilih secara acak kemudian ditukar. Gen akan dipilih dan diacak sebanyak $1 * \vartheta$ kali.

Kemudian masuk ke penghitungan fitness, hasil perhitungan fitness ditunjukkan oleh tabel 4.17.

Tabel 4.10 Nilai fitness c4

populasi	asal	gapplek	onggok	cantel	tetes	ϑ	fitness
c2	p1	1,7730	2,8850	2,5640	2,3230	2,2	0,2945

Karena λ bernilai 2 maka inisialisasi individu yang dihasilkan dari parent p1 berhenti dan fitness c2 tidak perlu dibandingkan dengan fitness p1, kemudian berlanjut untuk membuat individu baru dari p2 dengan nilai ϑ yang kembali menjadi random 1-3. Proses berlangsung terus hingga keseluruhan individu induk masing-masing memiliki 2 individu baru atau child.

Hasil dari iterasi ditunjukkan oleh tabel 4.18

Tabel 4.11 Hasil mutasi

populasi	asal	gapplek	onggok	cantel	tetes	ϑ	fitness
c1	p1	2,5640	2,3230	1,7730	2,8850	2	0,3260
c2	p1	1,7730	2,8850	2,5640	2,3230	2,2	0,2945
c3	p2	2,6560	2,7430	2,4550	2,1350	2	0,2886
c4	p2	2,6560	2,1350	2,7430	2,4550	1,8	0,2759
c5	p3	1,7670	2,2230	2,8870	2,2540	3	0,2881
c6	p3	1,7670	2,2230	2,2540	2,8870	2,7	0,3139
c7	p4	2,5470	1,9840	1,9700	2,4530	2	0,3305
c8	p4	2,5470	2,4530	1,9700	1,9840	1,8	0,3331
c9	p5	1,9060	2,6440	1,5560	1,8540	1	0,3909
c10	p5	2,6440	1,9060	1,8540	1,5560	1,1000	0,3664

Jika dikoreksi lagi bisa dilihat bahwa c1 adalah individu awal yang dibentuk dari p1 sehingga θ bernilai random 1-3 kemudian karena fitness c1 lebih dari fitness p1 maka untuk inisialisasi c2 θ bernilai 2,2. Kemudian untuk c3 karena parent berganti maka nilai θ di set lagi dari awal yaitu random 1-3 dan karena nilai fitness c3 kurang dari p2 maka untuk inisialisasi c4 nilai θ diturunkan menjadi 1,8.

4.2.4 Seleksi

Tahap seleksi adalah tahap untuk mengurangi jumlah popsize yang telah bertambah menjadi jumlah popsize semula, pada kasus ini jumlah popsize akan menjadi 5,. Pada penelitian ini seleksi menggunakan elitism yaitu keseluruhan kromosom akan diurutkan dari fitness terbesar hingga terkecil kemudian diambil kromosom sejumlah popsize awal. Hasil dari seleksi ditunjukkan oleh tabel 4.19.

Tabel 4.12 Hasil Seleksi

populasi	asal	gapplek	onggok	cantel	tetes	fitness
p1	c9	1,9060	2,6440	1,5560	1,8540	0,3909
p2	p5	1,906	2,644	1,854	1,556	0,3715
p3	c10	2,6440	1,9060	1,8540	1,5560	0,3664
p4	c8	2,5470	2,4530	1,9700	1,9840	0,3331
p5	p4	1,97	2,547	1,984	2,453	0,3329