

## BAB 5 HASIL DAN PEMBAHASAN

Bab ini membahas hasil serta pembahasan dari skenario pengujian yang telah diimplementasikan. Pengujian mekanisme hibridisasi membahas efektifitas di dalam model hibridisasi tersebut. Pengujian parameter *maximum probability* ditujukan untuk menentukan parameter yang baik terhadap pergerakan partikel menggunakan metode biseksi. Pengujian jumlah *sub-swarm* dilakukan untuk mengetahui hubungan jumlah *sub-swarm* dengan nilai *fitness* yang dihasilkan. Pengujian *swarm size*, iterasi, koefisien akselerasi, dan *inertia weight* ditujukan untuk menentukan parameter yang baik sedangkan pengujian komparasi dilakukan untuk menguji seberapa besar tingkat efektifitas model hibridisasi yang diajukan dibandingkan dengan metode komputasi evolusioner lainnya.

### 5.1 Mekanisme Hibridisasi

Mekanisme hibridisasi diuji untuk mengetahui pengaruh rata-rata *fitness* terhadap skenario mekanisme yang berbeda dan menentukan mekanisme yang terbaik. Hasil skenario pengujian pada 4 jenis mekanisme hibridisasi dari 5 formula yang berebeda dapat dilihat pada Tabel 5.3.

**Tabel 5.1 Hasil skenario pengujian mekanisme hibridisasi**

Formula	Mekanisme Hibridisasi			
	A	B	C	D
A11	<b>0,363071197</b>	0,362858612	0,361562809	0,356975521
A12	0,406933312	0,407044663	<b>0,407225012</b>	0,406708561
A13	0,567041262	<b>0,586745068</b>	0,55689411	0,551181424
A14	1,272392106	1,274505469	<b>1,27631503</b>	1,267415963
A15	1,137145194	<b>1,149077408</b>	1,133458968	1,135152375
Rata-Rata	0,749316614	<b>0,756046244</b>	0,747091186	0,743486769

Tulisan dengan cetak tebal pada Tabel 5.3 menunjukkan rata-rata nilai *fitness* tertinggi. Pada formula A11, A yaitu mekanisme dimana terdapat *sub-swarm* yang memiliki parameter maksimum probabilitas biseksi sebesar 0, menghasilkan nilai *fitness* tertinggi dibandingkan mekanisme hibridisasi yang lain. Pada formula A12 dan A14, C yaitu model hibridisasi tanpa melibatkan fase agregasi dapat menghasilkan nilai *fitness* tertinggi tetapi peningkatan tidak terlalu signifikan. Sedangkan pada formula A13 dan A15, B yaitu model hibridisasi dengan tanpa melibatkan nilai maksimum probabilitas biseksi sebesar 0 dapat menghasilkan nilai *fitness* tertinggi. Hasil rata-rata dari semua formula menunjukkan mekanisme hibridisasi B merupakan mekanisme yang terbaik dengan dihasilkan rata-rata tertinggi. Sehingga, B menjadi mekanisme hibridisasi yang akan digunakan selanjutnya dan pengujian parameter maksimum probabilitas biseksi akan diuji tanpa melibatkan parameter dengan nilai 0. Pada pengujian ini digunakan parameter  $w = 0,6$ ,  $c1 = 1,8$ ,  $c2 = 2.1$ , dan iterasi = 5,000

## 5.2 Parameter Maksimum Probabilitas Biseksi

Karena mekanisme hibridisasi dengan model B, parameter maximum probability (MaxBp) diatas 0 untuk semua *sub-swarm*, menunjukkan rata-rata tertinggi dibandingkan mekanisme yang lain maka rentang nilai parameter MaxBp antara 0,1 sampai 1,0 diuji dengan kenaikan 0,1. Pada pengujian ini digunakan parameter  $w = 0,6$ ,  $c1 = 1,8$ ,  $c2 = 2.1$ , dan iterasi = 5,000. Hasil pengujian parameter MaxBp dapat dilihat pada Tabel 5.4

**Tabel 5.2 Parameter MaxBp optimal dalam menghasilkan rata-rata nilai fitness terbaik**

Formula	Rata-Rata Fitness	Parameter MaxBp Optimal
A11	0,364532727	0,4
A12	0,407134592	0,5
A13	0,594447983	0,3
A14	1,289495266	0,5
A15	1,154269806	0,2

Hasil yang diperoleh menunjukkan rata-rata *fitness* terbaik memerlukan nilai parameter MaxBp yang berbeda-beda seperti terlihat pada Tabel 5.4. Penentuan parameter tersebut berdasarkan tingkat signifikan perubahan pada nilai *fitness*. Nilai MaxBp dipilih jika nilai MaxBp berikutnya tidak menghasilkan perubahan yang signifikan atau fluktuatif tetapi tidak menunjukkan peningkatan yang signifikan.

Diperlukannya nilai parameter MaxBp yang berbeda sangat dimungkinkan dipengaruhi oleh kompleksitas ruang pencarian yang berbeda-beda tergantung dari kombinasi bahan pakan dan kebutuhan zat makanan. Oleh sebab itu, menentukan parameter yang terbaik untuk semua kemungkinan kombinasi bahan pakan akan menjadi sangat sulit mengingat ketersediaan bahan pakan, harga yang fluktuatif, dan kebutuhan nutrisi yang berbeda-beda. Strategi yang sebaiknya digunakan dalam menentukan parameter MaxBp adalah dengan tidak mengurangi kualitas solusi secara signifikan. Sehingga, nilai MaxBp yang berbeda-beda ditetapkan pada tiap sub-swarm. Nilai parameter yang digunakan adalah 0,2, 0,3, 0,4, dan 0,5 berdasarkan parameter yang optimal yang ditemukan pada Tabel 5.4.

**Tabel 5.3 Perbandingan Parameter MaxBp Gabungan dengan Parameter MaxBp Optimal**

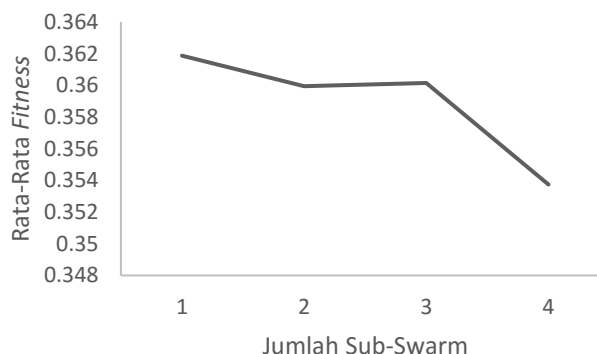
Formula	MaxBp				Gabungan
	0.2	0.3	0.4	0.5	
A11	0,361087193	0,359166568	<b>0,364532727</b>	0,359213485	0,363114584
A12	0,406755297	0,406908529	0,407074427	<b>0,407134592</b>	0,407080893
A13	0,56698449	<b>0,594447983</b>	0,580853844	0,586662254	0,579360744
A14	1,257855129	1,27840184	1,283659853	<b>1,289495266</b>	1,289408224

A15	<b>1,154269806</b>	1,147400843	1,145265406	1,147903396	<i>1,14920834</i>
Rata-Rata	0,749390383	0,757265153	0,756277251	<b>0,758081798</b>	<b>0,757634557</b>

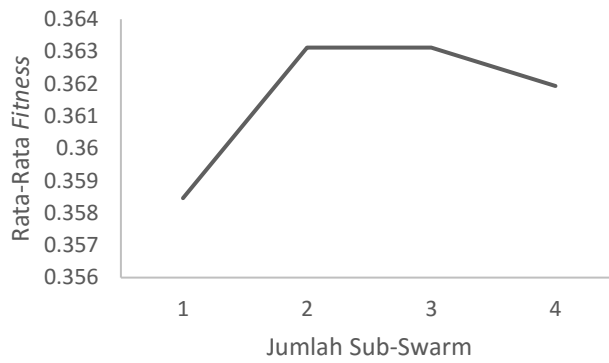
Hasil dari uji coba gabungan nilai parameter MaxBp dapat dilihat pada Tabel 5.5. Nilai dengan cetak tebal menunjukkan rata-rata *fitness* optimum dari parameter MaxBp optimum dan terbaik dibandingkan dengan parameter MaxBp yang lain. Lebih rinci setiap formula memerlukan nilai MaxBp yang berbeda-beda. Seperti pada Formula A11 diperoleh nilai 0,4 sebagai parameter optimum dan memberikan hasil yang lebih baik dibandingkan nilai MaxBp dibawah atau diatasnya. Hasil rata-rata *fitness* dari parameter MaxBp gabungan dapat memberikan nilai yang baik dimana lebih rendah dibandingkan dengan rata-rata *fitness* terbaik tetapi lebih tinggi dibandingkan dengan hasil rata-rata *fitness* lainnya yang ditandakan dengan cetak miring. Pada formula A11, nilai rata-rata *fitness* gabungan adalah 0,363114584 yang lebih rendah dibandingkan dengan rata-rata *fitness* dari MaxBp optimum yaitu sebesar 0,364532727 tetapi lebih tinggi dibandingkan dengan rata-rata *fitness* lainnya yaitu sebesar 0,361087193, 0,359166568, dan 0,359213485. 4 dari 5 rata-rata *fitness* dari parameter MaxBp gabungan dapat memberikan solusi yang baik. Sehingga parameter MaxBp gabungan dapat dijadikan sebagai parameter yang baik dalam menghasilkan solusi formula yang optimum.

### 5.3 Jumlah *Sub-Swarm*

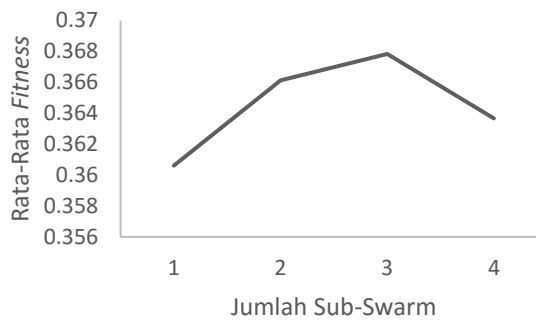
Kenaikan jumlah *sub-swarm* diuji dengan *swarm size* yang sama. Hasil uji jumlah *sub-swarm* dengan *swarm size* yang sama untuk A11 sampai A15 dapat dilihat pada Gambar 5.1 sampai Gambar 5.5. Pengujian jumlah *sub-swarm* menggunakan MaxBp yang diperoleh dari hasil pengujian *maximum probability* yang telah dibahas pada sub-bab 5.2 dengan parameter  $w = 0,6$ ,  $c_1 = 1,8$ ,  $c_2 = 2.1$ , dan iterasi = 5,000.



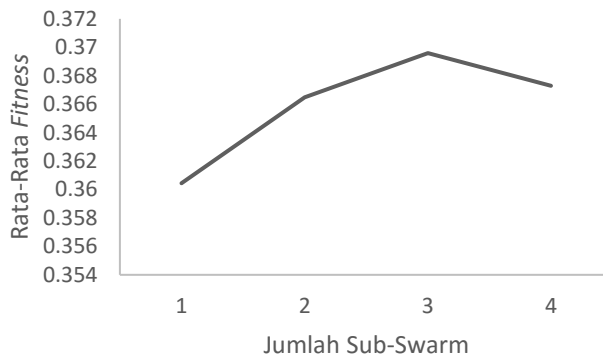
**Gambar 5.1 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 60 untuk A11**



**Gambar 5.2 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 120 untuk A11**



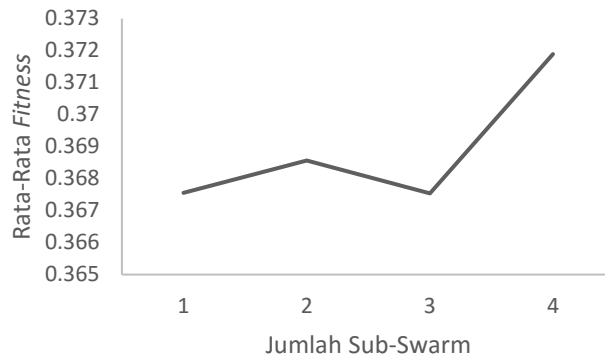
**Gambar 5.3 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 180 untuk A11**



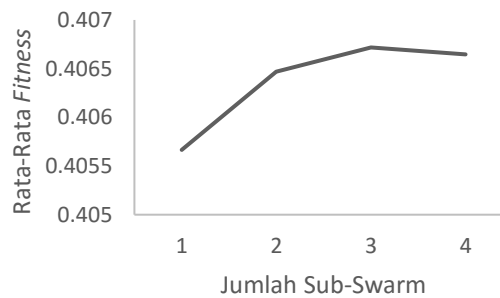
**Gambar 5.4 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 240 untuk A11**

*Swarm size* sebesar 60 dapat memberikan rata-rata *fitness* yang baik namun terus mengalami penurunan jika jumlah tersebut dibagi menjadi beberapa *sub-swarm* seperti terlihat pada Gambar 5.1. *Swarm size* yang kecil pada tiap-tiap *sub-swarm* memungkinkan terjadinya konvergensi dini karena kurangnya keberagaman sehingga berkontribusi pada penurunan rata-rata *fitness*. Jumlah *sub-swarm* sebesar 4 dengan masing-masing *swarm size* sebesar 15 tidak dapat memberikan solusi yang lebih baik dibandingkan satu *swarm* dengan *swarm size*

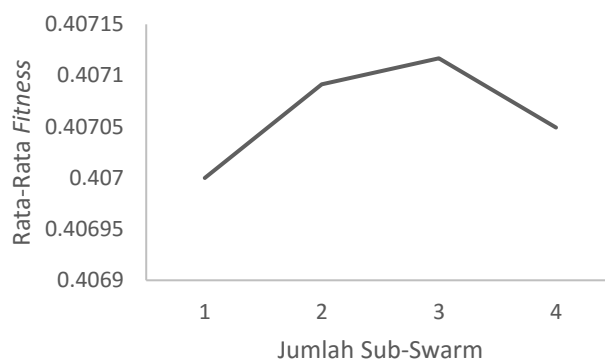
sebesar 60. Namun pada *swarm size* sebesar 120 dua *sub-swarm* dapat memberikan rata-rata *fitness* lebih tinggi dibandingkan hanya satu *swarm* saja seperti terlihat pada Gambar 5.2. Lebih jauh lagi, pada gambar 5.3 sampai 5.4 rata-rata nilai *fitness* dapat ditingkatkan dengan memanfaatkan *sub-swarm* lebih dari satu.



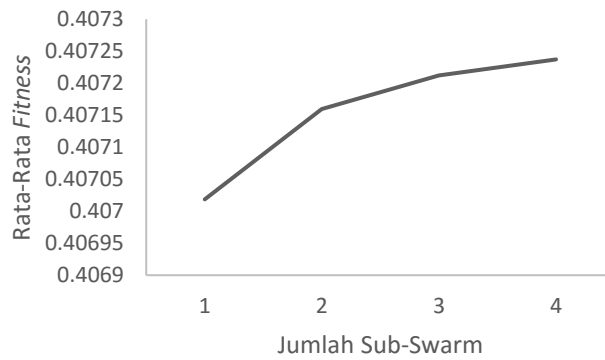
**Gambar 5.5 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 300 untuk A11**



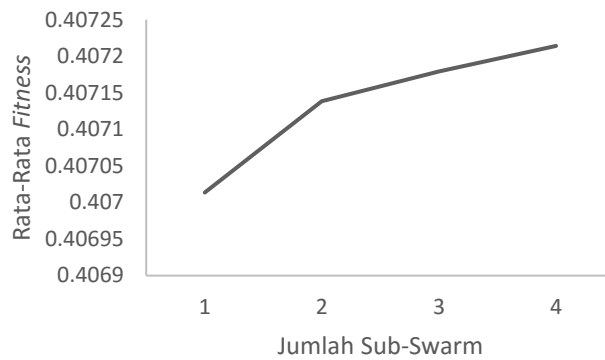
**Gambar 5.6 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 60 untuk A12**



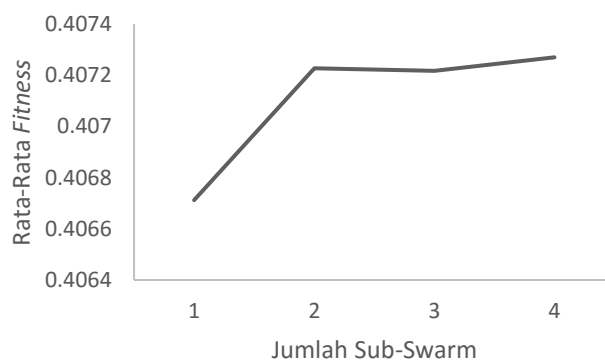
**Gambar 5.7 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 120 untuk A12**



**Gambar 5.8 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 180 untuk A12**



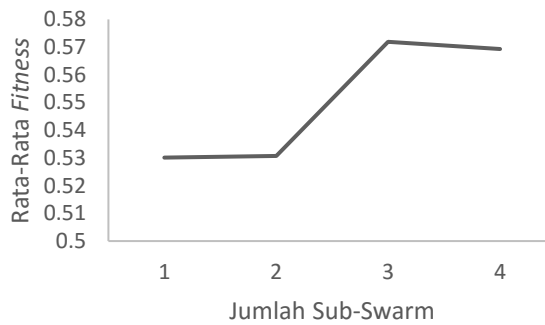
**Gambar 5.9 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 240 untuk A12**



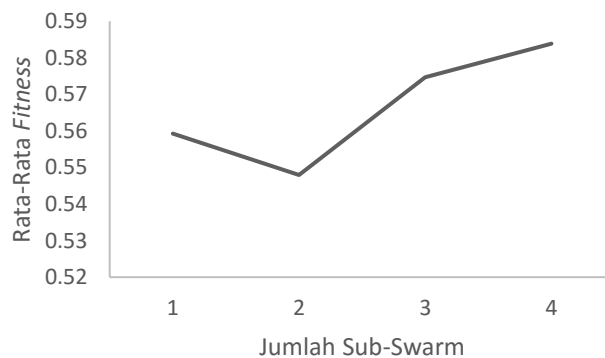
**Gambar 5.10 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 300 untuk A12**

Dengan menggunakan parameter yang sama, kenaikan jumlah sub-swarm diuji untuk formula A12. Hasil pengujian tersebut dapat dilihat pada Gambar 5.6 – 5.10. Hasil rata-rata *fitness* untuk A12 menunjukkan bahwa jumlah *sub-swarm* sensitif

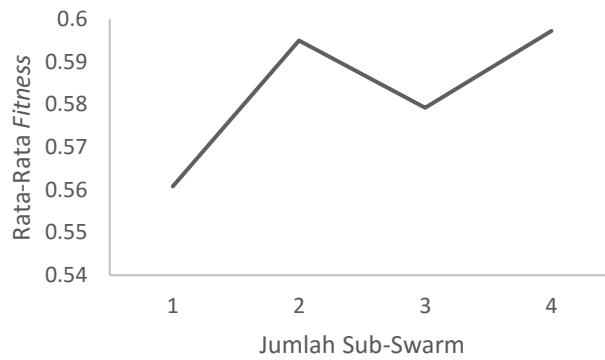
dengan rata-rata fitness mulai dari *swarm size* sebesar 60 seperti terlihat pada Gambar 5.6. Namun satu *swarm* untuk *swarm size* sebesar 120 dapat memberikan rata-rata *fitness* lebih tinggi dibandingkan dengan *swarm size* sebesar 60 seperti terlihat pada Gambar 6.7. Ini menunjukkan bahwa 60 merupakan ukuran yang belum optimal dalam formula A12. Pada Gambar 6.7, rata-rata *fitness* dapat ditingkatkan seiring dengan banyaknya jumlah *sub-swarm* namun mengalami penurunan pada jumlah *sub-swarm* sebesar 4 yang memiliki *swarm size* sebesar 30 untuk tiap-tiap *sub-swarm*. Ini menunjukkan bahwa semakin banyak jumlah *sub-swarm* tidak menjamin semakin meningkat rata-rata fitness atau memberikan solusi yang lebih baik. Hal ini mungkin terjadi jika *swarm size* berukuran kecil untuk tiap-tiap *sub-swarm* sehingga menyebabkan terjebak dinilai optimum lokal karena kurangnya keberagaman dalam *sub-swarm* itu sendiri. Sebaliknya *swarm size* dalam ukuran lebih besar dapat memberikan rata-rata *fitness* yang lebih tinggi seiring bertambahnya jumlah *sub-swarm* seperti terlihat pada Gambar 5.8 sampai 5.10.



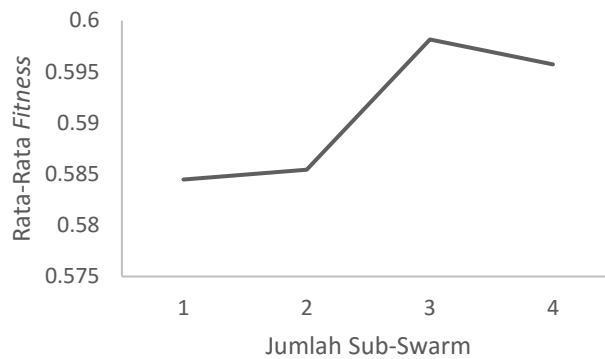
**Gambar 5.11 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 60 untuk A13**



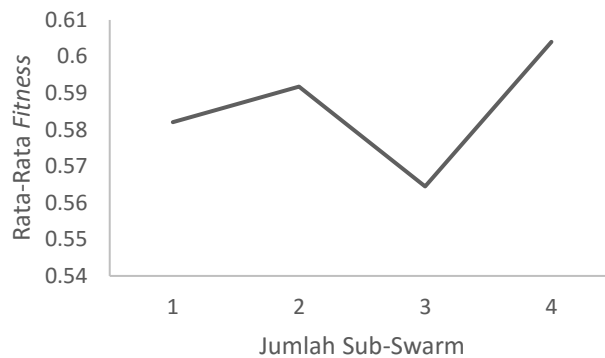
**Gambar 5.12 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 120 untuk A13**



**Gambar 5.13 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 180 untuk A13**



**Gambar 5.14 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 240 untuk A13**

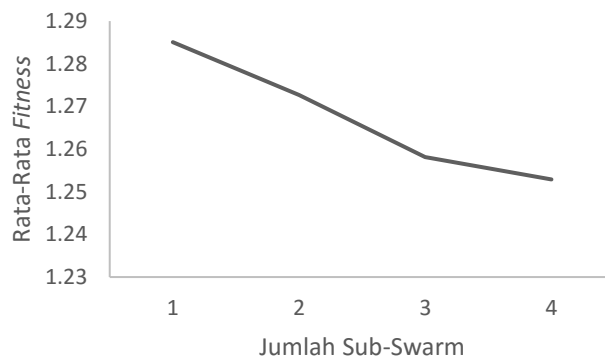


**Gambar 5.15 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 300 untuk A13**

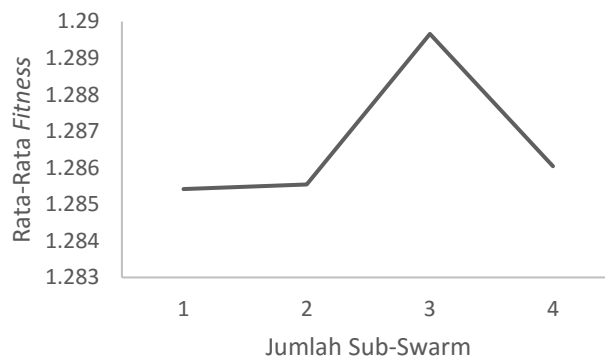
Pada A13 rata-rata *fitness* terus meningkat secara fluktuatif dengan pertambahan jumlah *sub-swarm*. Pada Gambar 5.11, rata-rata *fitness* terus meningkat dengan *swarm size* sebesar 60. Namun *fitness* yang dihasilkan dapat dikatakan belum optimal dikarenakan jumlah *swarm size* sebesar 120 dalam satu



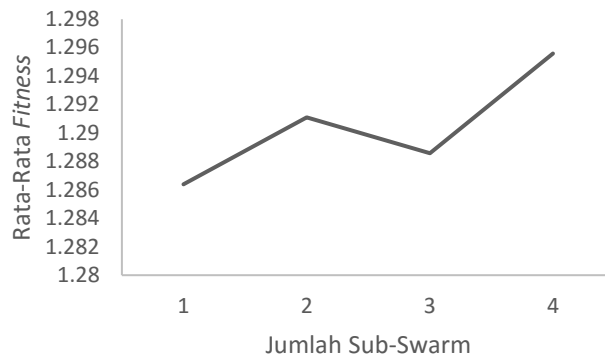
*swarm* dapat memberikan rata-rata *fitness* yang lebih tinggi seperti terlihat pada Gambar 5.12. Dua *sub-swarm* menghasilkan rata-rata *fitness* yang lebih rendah dibandingkan dengan satu *sub-swarm* saja namun mengalami peningkatan pada 3 dan 4 *sub-swarm*. Pada kasus seperti ini dimungkinkan ruang pencarian lebih kompleks dibandingkan dengan dua formula yang telah dibahas sebelumnya sehingga menyebabkan partikel sulit untuk keluar dari nilai optimum lokal yang juga dipengaruhi oleh bangkitan nilai random pada komponen sosial dan kognitif pada PSO. Hasil rata-rata fluktuatif ini juga ditemukan pada hasil pengujian untuk *swarm size* sebesar 280, 240, dan 300 seperti terlihat pada Gambar 5.13, 5.14 dan 5.15. Walaupun rata-rata *fitness* fluktuatif namun dapat dilihat bahwa adanya peningkatan rata-rata *fitness* dengan bertambahnya *sub-swarm*.



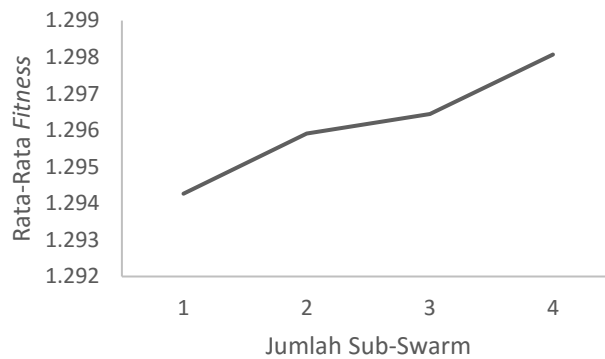
**Gambar 5.16 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 60 untuk A14**



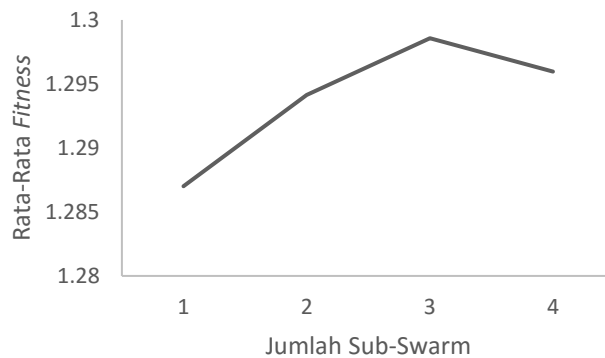
**Gambar 5.17 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 120 untuk A14**



**Gambar 5.18 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 180 untuk A14**



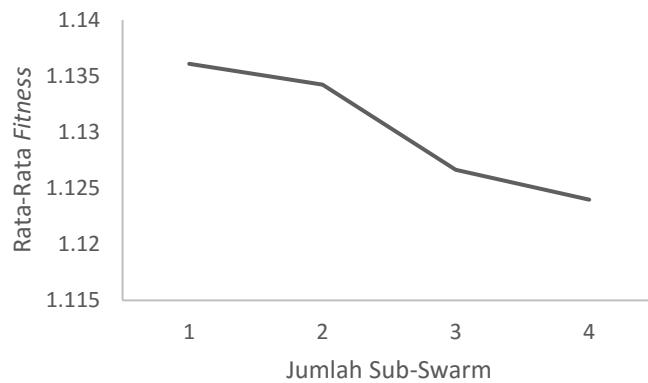
**Gambar 5.19 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 240 untuk A14**



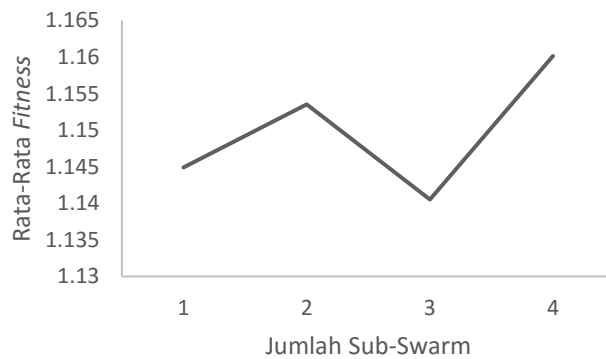
**Gambar 5.20 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 300 untuk A14**

Hasil pengujian *sub-swarm* untuk A14 dapat dilihat pada Gambar 5.16 sampai Gambar 5.20. Pola penurunan rata-rata *fitness* ditemukan pada *swarm size* sebesar 60 yang juga ditemukan pada A11 seperti terlihat pada Gambar 5.16. Pada *swarm size* sebesar 120 seperti terlihat pada Gambar 5.17, rata-rata *fitness*

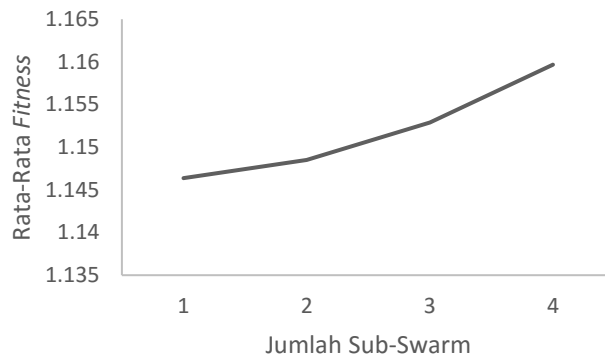
mengalami peningkatan dari 1 sampai 3 *sub-swarm* namun mengalami penurunan yang cukup signifikan dengan jumlah *sub-swarm* sebesar 4. Sama halnya yang ditemukan pada A11 dan A12, kurangnya keberagaman dapat memicu konvergensi dini pada nilai optimum lokal yang lebih rendah. Sedangkan pada Gambar 5.18, pola tersebut memiliki kesamaan pada A13 yang menghasilkan rata-rata *fitness* yang fluktuatif namun mengalami peningkatan dengan menggunakan 4 *sub-swarm*. Pada Gambar 5.19, rata-rata *fitness* terlihat sensitif dengan jumlah *sub-swarm*. Sedangkan pada Gambar 5.10, rata-rata *fitness* juga terus meningkat namun adanya penurunan yang tidak signifikan pada 4 *sub-swarm*.



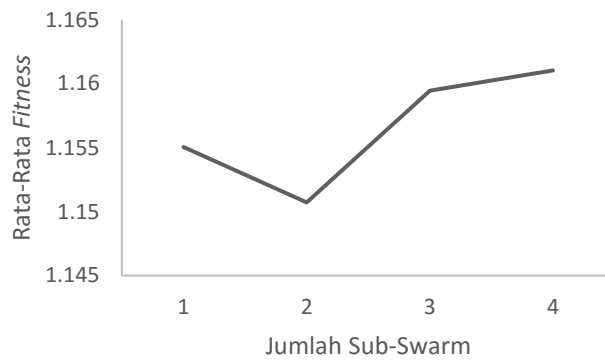
**Gambar 5.21 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 60 untuk A15**



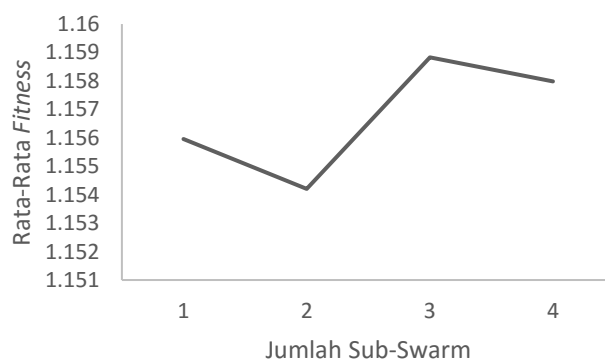
**Gambar 5.22 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 120 untuk A15**



**Gambar 5.23 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 180 untuk A15**



**Gambar 5.24 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 240 untuk A15**



**Gambar 5.25 Rata-Rata *fitness* dari jumlah *sub-swarm* 1 sampai 4 dengan jumlah *swarm size* sebesar 300 untuk A15**

Seperti halnya pada A11 dan A14, dengan *swarm size* sebesar 60, nilai *fitness* mengalami penurunan seiring bertambahnya *sub-swarm* untuk formula A15 seperti terlihat pada Gambar 5.21. Namun rata-rata *fitness* secara fluktuatif

meningkat seiring bertambahnya *sub-swarm* dengan *swarm size* sebesar 120 seperti terlihat pada Gambar 5.22. Dengan *swarm size* sebesar 180, penambahan *sub-swarm* memberikan dampak positif secara bertahap pada rata-rata *fitness* yang dihasilkan seperti terlihat pada Gambar 5.23. Sedangkan pada Gambar 5.24 dan 5.24 fluktuasi rata-rata *fitness* terus meningkat seiring bertambahnya *sub-swarm*.

Dari semua hasil pengujian jumlah *sub-swarm* untuk 5 bahan uji yang berbeda (A11 – A15), pola rata-rata *fitness* terhadap jumlah *sub-swarm* antara 1 sampai 4 sangat dipengaruhi pada kombinasi bahan pakan yang digunakan yang berkontribusi pada kompleksitas ruang pencarian. 3 dari 5 bahan uji (A11, A14, dan A15) *swarm size* sebesar 60 dapat menurunkan nilai *fitness* secara bertahap dengan bertambahnya jumlah *sub-swarm*. Ketika 4 *sub-swarm* digunakan, maka masing-masing *sub-swarm* memiliki *swarm size* sebesar 15 yang jauh berbeda jika dibandingkan satu *sub-swarm* dengan *swarm size* sebesar 60. Kecilnya *swarm size* pada *sub-swarm* akan mengurangi tingkat keberagaman dibandingkan hanya satu *swarm* saja. Dapat dikatakan bahwa 60 *swarm size* untuk beberapa kasus memiliki keberagaman yang lebih baik dibandingkan 15 *swarm size* untuk 4 *sub-swarm* sehingga berkontribusi secara signifikan terhadap penurunan rata-rata *fitness* namun penurunan tersebut tidak ditemukan pada bahan uji A12 dan A13. Sehingga walaupun total *swarm size* cukup tinggi tetapi *swarm size* untuk tiap-tiap *sub-swarm* kecil dapat memperburuk performa PSO yang memanfaatkan strategi *multi-swarm*.

Tidak semua hasil uji menunjukkan kenaikan yang bertahap dengan bertambahnya jumlah *sub-swarm*. Pada bahan uji A11, A13, A14, dan A15 terdapat pola yang fluktuatif dari rata-rata *fitness* yang dihasilkan namun secara bertahap rata-rata *fitness* dapat ditingkatkan dengan bertambahnya jumlah *sub-swarm*. Hal ini menunjukkan bahwa sangat memungkinkan ketika 3 *sub-swarm* digunakan dapat memberikan performa yang lebih buruk dibandingkan 2 *sub-swarm*. Namun ketika hibridisasi PSO tersebut dijalankan kembali sangat memungkinkan terjadi hal yang sebaliknya dimana 3 *sub-swarm* memberikan hasil yang lebih baik dibandingkan 2 *sub-swarm*. Tidak semua kasus hal tersebut terjadi, pada bahan uji A12 dengan jumlah *swarm size* yang optimal, seiring bertambahnya jumlah *sub-swarm*, rata-rata *fitness* dapat ditingkatkan secara bertahap. Tetapi dari semua hasil uji jumlah *sub-swarm* dengan *swarm size* yang optimal untuk tiap-tiap *sub-swarm*, 4 *sub-swarm* selalu menunjukkan hasil yang lebih baik dibandingkan dengan 2 *sub-swarm*. Hasil uji tersebut menunjukkan bahwa dengan memanfaatkan banyaknya *sub-swarm* pada PSO akan meningkatkan peluang partikel dalam menemukan solusi yang lebih baik dengan total *swarm size* yang sama. Oleh sebab itu, strategi ini lebih baik dibandingkan dengan standar PSO.

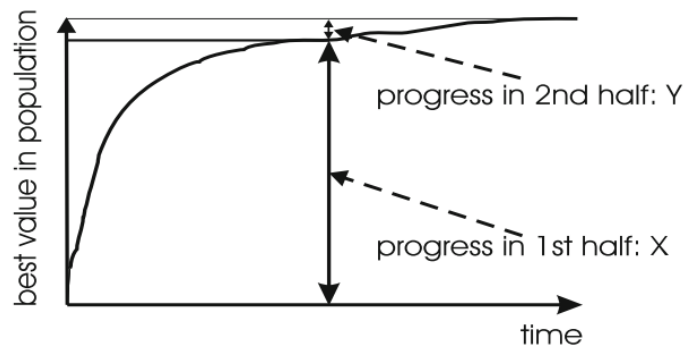
Dari hasil uji, dampak peningkatan jumlah *sub-swarm* berbeda-beda pada tiap bahan uji. Namun semuanya menunjukkan adanya peningkatan dengan meningkatnya jumlah *sub-swarm* ketika *swarm size* ditingkatkan. Adanya peningkatan seiring dengan bertambahnya jumlah *sub-swarm* juga sejalan dengan studi Gonsalves dan Egashira (2013). Namun pengujian yang mereka lakukan

meningkatkan jumlah *swarm size* seiring dengan peningkatan jumlah *sub-swarm* yang menyebabkan kurang adilnya pengujian dikarenakan jumlah *swarm size* pada setiap peningkatan *sub-swarm* tidak sama.

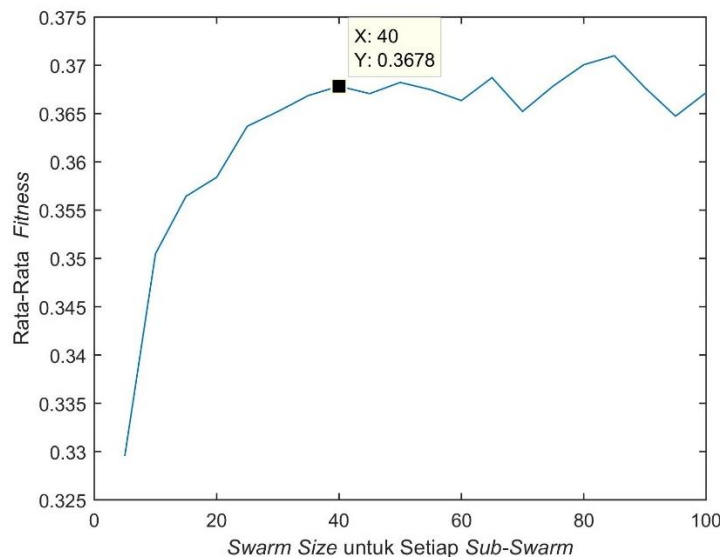
Sepengetahuan penulis, tidak ada penulis lain yang membahas bagaimana menentukan jumlah *sub-swarm* yang optimum. Sehingga jumlah *sub-swarm* yang baik ditentukan berdasarkan hasil uji tersebut dan 4 *sub-swarm* ditentukan sebagai jumlah yang baik.

### 5.4 Swarm Size

Pengujian *swarm size* dilakukan dengan menaikkan nilai tersebut secara bertahap terhadap 5 formula yang berbeda. Lalu *swarm size* yang optimum dari masing-masing formula ditentukan dan parameter yang baik dari semua formula juga ditentukan untuk parameter MSPSO.



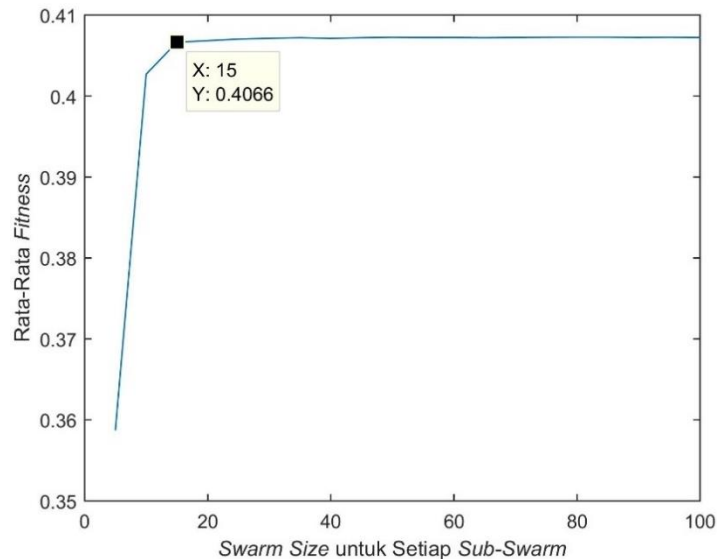
**Gambar 5.26** Ketidakefektifan peningkatan waktu komputasi terhadap *fitness*. Area X menunjukkan kenaikan *fitness* pada setengah pertama algoritme dijalankan dan area Y setengah lainnya (Eiben and Smith, 2015).



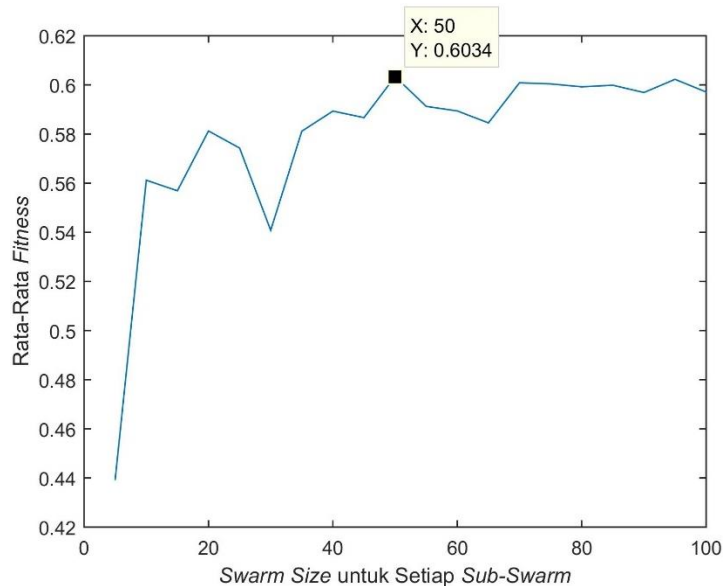
**Gambar 5.27** Rata-rata *fitness* untuk *swarm size* yang berbeda pada A11

Penentuan *swarm size* yang optimum adalah dengan memperhatikan tingkat signifikan perubahan pada rata-rata *fitness*. Jika rata-rata *fitness* tidak meningkat

signifikan setelah kenaikan *swarm size* berikutnya, maka pemilihan *swarm size* yang optimum pada area tersebut akan dihindari. Pendekatan ini juga digunakan dalam algoritme evolusi dengan membagi dua bagian, yaitu bagian yang terus mengalami peningkatan (area X) dan bagian yang tidak mengalami peningkatan yang signifikan (area Y). Sehingga walaupun ada peningkatan pada area Y, pemilihan parameter tersebut sebaiknya dihindari karena peningkatan yang diberikan tidak menjanjikan dan akan meningkatkan waktu komputasi (Eiben and Smith, 2015). Lebih lanjut, area tersebut tergambar pada Gambar 5.26.



**Gambar 5.28 Rata-rata *fitness* untuk *swarm size* yang berbeda pada A12**

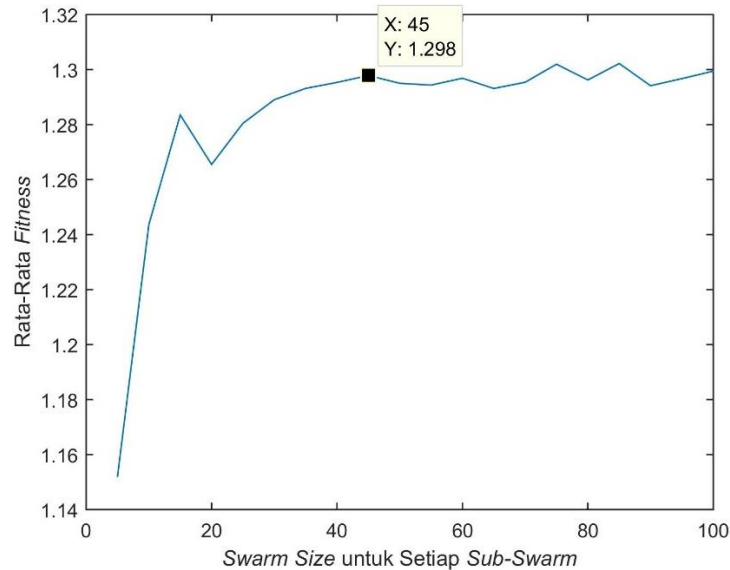


**Gambar 5.29 Rata-rata *fitness* untuk *swarm size* yang berbeda pada A13**

Hasil pengujian *swarm size* dalam menemukan parameter yang optimum dari A11 sampai A15 dapat dilihat pada Gambar 5.27 sampai 5.31. Pengujian ini menggunakan MaxBp dari hasil pengujian parameter *maximum probability* yang telah dibahas pada sub-bab 5.2 dan menggunakan jumlah *sub-swarm* yang baik

yang telah dibahas pada sub-bab 5.3 dengan parameter  $w = 0,6$ ,  $c_1 = 1,8$ ,  $c_2 = 2.1$ , dan iterasi = 10,000.

Hasil rata-rata *fitness* terhadap bahan uji A11 menunjukkan adanya kenaikan setiap penambahan *swarm size* seperti terlihat pada Gambar 5.27. Secara bertahap rata-rata *fitness* meningkat dengan *swarm size* mulai dari 5 sampai dengan 40. Namun rata-rata *fitness* tidak meningkat secara signifikan setelah 40 *swarm size*. Sehingga pada kasus A11, 40 menjadi *swarm size* yang optimum untuk tiap-tiap *sub-swarm* dengan total *swarm size* sebesar 160.



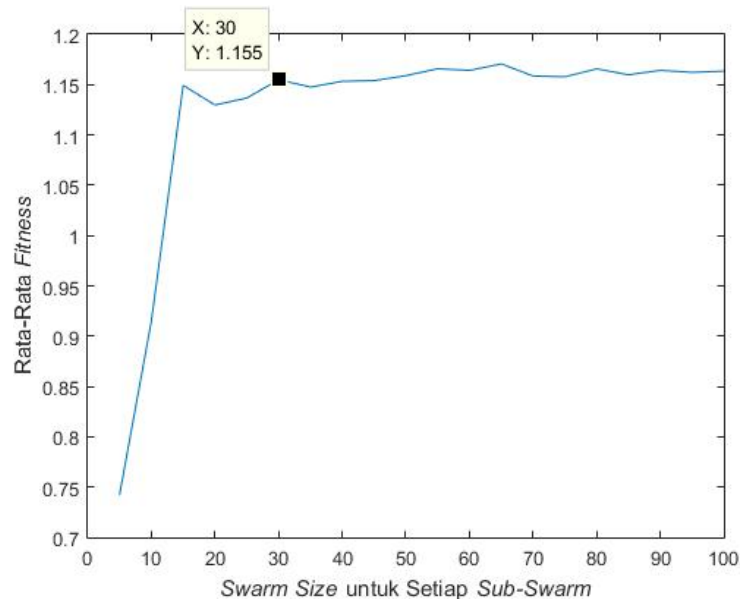
**Gambar 5.30 Rata-rata *fitness* untuk *swarm size* yang berbeda pada A14**

Hasil uji *swarm size* pada bahan uji A12 dapat dilihat pada Gambar 5.28. Rata-rata *fitness* meningkat signifikan mulai 5 sampai 15 *swarm size*. Setelah 15 *swarm size*, rata-rata *fitness* tidak mengalami peningkatan yang signifikan. Sehingga 15 menjadi *swarm size* yang optimum pada A12 dengan total *swarm size* sebesar 60.

Hasil uji *swarm size* pada bahan uji A13 dapat dilihat pada Gambar 5.29. Rata-rata *fitness* meningkat signifikan mulai 5 sampai 50 *swarm size*. Setelah 50 *swarm size*, rata-rata *fitness* tidak mengalami peningkatan yang signifikan. Sehingga 50 menjadi *swarm size* yang optimum pada A13 dengan total *swarm size* sebesar 200.

Hasil uji *swarm size* pada bahan uji A14 dapat dilihat pada Gambar 5.30. Rata-rata *fitness* meningkat signifikan mulai 5 sampai 45 *swarm size*. Setelah 45 *swarm size*, rata-rata *fitness* tidak mengalami peningkatan yang signifikan. Sehingga 45 menjadi *swarm size* yang optimum pada A14 dengan total *swarm size* sebesar 180.





**Gambar 5.31 Rata-rata *fitness* untuk *swarm size* yang berbeda pada A15**

Hasi uji *swarm size* pada bahan uji A15 dapat dilihat pada Gambar 5.31. Rata-rata *fitness* meningkat signifikan mulai 5 sampai 30 *swarm size*. Setelah 30 *swarm size*, rata-rata *fitness* tidak mengalami peningkatan yang signifikan. Sehingga 30 menjadi *swarm size* yang optimum pada A14 dengan total *swarm size* sebesar 120.

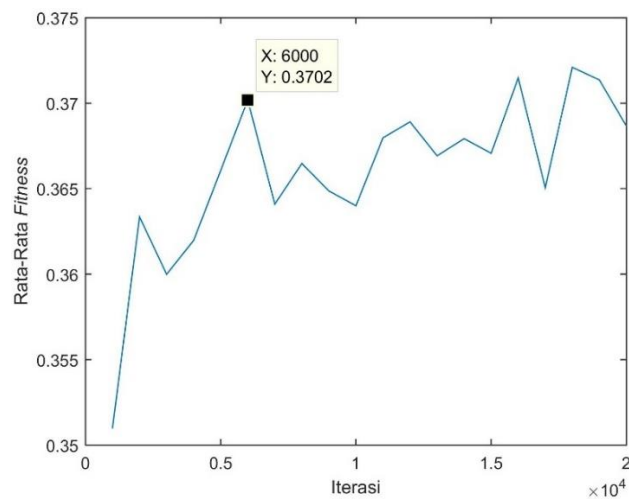
Semua optimum *swarm size* dari bahan uji yang berbeda-beda yang diperoleh dari skenario pengujian *swarm size* menunjukkan bahwa setiap bahan uji memerlukan parameter *swarm size* yang berbeda-beda untuk dapat menghasilkan solusi yang baik. Pada bahan uji A11, A12, A13, A14 dan A15 masing-masing memerlukan *swarm size* minimal berturut-turut sebesar 40, 15, 50, 45, dan 30 untuk menghasilkan solusi yang optimum. Menentukan *swarm size* yang optimal pada permasalahan optimasi pakan ternak diperlukan penyesuaian terhadap kombinasi bahan pakan itu sendiri yang terdiri dari nutrisi dan harga. Selain itu kebutuhan nutrisi juga dapat mempengaruhi kompleksitas ruang pencarian. Sehingga tidak ada *swarm size* yang optimal untuk semua kasus kombinasi bahan pakan yang akan diformulasikan menggunakan hibridisasi PSO. Namun, *swarm size* perlu ditentukan agar hibridisasi PSO dapat menghasilkan solusi yang baik walaupun parameter tersebut bukan yang paling optimum.

Karena *swarm size* yang terbaik sangat tergantung pada fitur suatu masalah seperti banyaknya dimensi, kendala, dan karakteristik dari suatu fungsi objektif (Rohler and Chen, 2011) dalam kasus ini yaitu jumlah kombinasi bahan pakan, kebutuhan zat makanan, dan fungsi *fitness* yang telah didefinisikan, maka akan menjadi tugas yang sangat sulit untuk menentukan semua nilai parameter yang terbaik. Namun, *swarm size* yang baik dapat ditentukan berdasarkan hasil uji *swarm size* yang diperoleh. Berdasarkan semua hasil *swarm size* yang optimal, 50 *swarm size* menjadi pilihan yang baik sebagai *swarm size* yang baik. Hal tersebut dikarenakan 50 *swarm size* dapat memberikan hasil yang baik pada kasus bahan uji lainnya yang memiliki nilai terbanyak diantara *swarm size* yang lain. Dengan menggunakan nilai parameter tersebut, rata-rata *fitness* yang diperoleh tidak

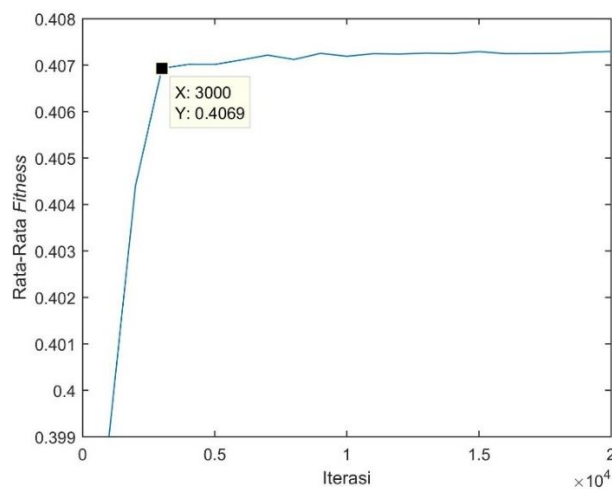
mengalami penurunan pada semua formula walaupun harus mengorbankan waktu komputasi pada beberapa formula. Dalam banyak kasus peningkatan *swarm size* berdampak pada pengurangan iterasi (Trelea, 2003). Sehingga 50 ditetapkan sebagai *swarm size* yang baik dalam optimasi formulasi pakan ayam petelur menggunakan hibridisasi PSO.

### 5.5 Jumlah Iterasi

Hasil pengujian jumlah iterasi dalam menemukan parameter yang optimum dari A11 sampai A15 dapat dilihat pada Gambar 5.32 sampai 5.36. Pendekatan yang digunakan dalam menentukan jumlah iterasi yang optimum sama dengan penentuan *swarm size* yang optimum yaitu dengan memperhatikan tingkat kenaikan yang signifikan terhadap jumlah iterasi. Pengujian ini menggunakan MaxBp dan *swarm size* dari hasil pengujian parameter *maximum probability* dan *swarm size* yang telah dibahas pada sub-bab 5.2 dan 5.4.



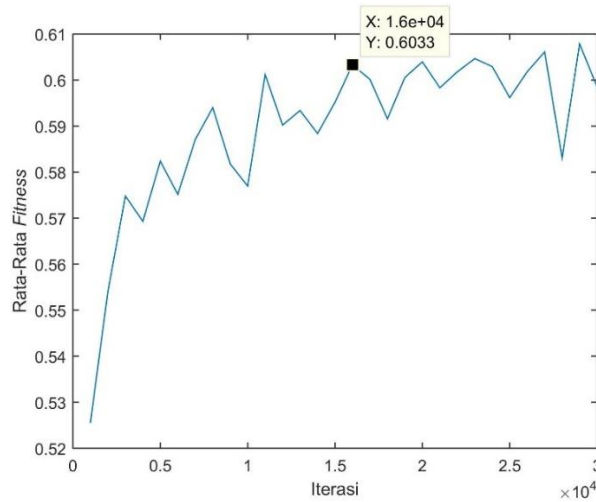
Gambar 5.32 Rata-rata *fitness* untuk jumlah iterasi yang berbeda pada A11



Gambar 5.33 Rata-rata *fitness* untuk jumlah iterasi yang berbeda pada A12

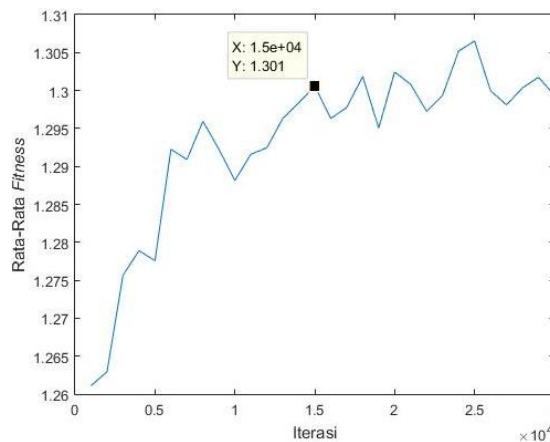
Hasil rata-rata *fitness* terhadap jumlah iterasi pada bahan uji A11 dapat dilihat pada Gambar 5.32. Rata-rata *fitness* secara bertahap dan fluktuatif terus meningkat mulai dari jumlah iterasi sebesar 1.000 sampai dengan 6.000. Namun rata-rata *fitness* tidak mengalami kenaikan setelah 6.000 iterasi. Sehingga, jumlah iterasi sebesar 6.000 menjadi jumlah iterasi yang optimum untuk bahan uji A11.

Hasil rata-rata *fitness* terhadap jumlah iterasi pada bahan uji A12 dapat dilihat pada Gambar 5.33. Rata-rata *fitness* secara bertahap terus meningkat mulai dari jumlah iterasi sebesar 1.000 sampai dengan 3.000. Namun rata-rata *fitness* tidak mengalami kenaikan setelah 3.000 iterasi. Sehingga, jumlah iterasi sebesar 3.000 menjadi jumlah iterasi yang optimum untuk bahan uji A12.

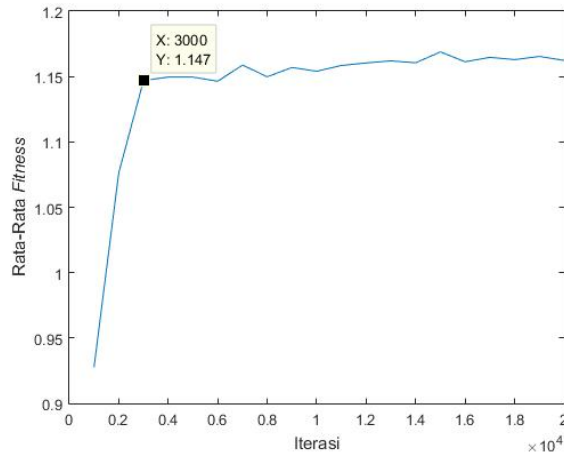


**Gambar 5.34 Rata-rata *fitness* untuk jumlah iterasi yang berbeda pada A13**

Hasil rata-rata *fitness* terhadap jumlah iterasi pada bahan uji A13 dapat dilihat pada Gambar 5.34. Rata-rata *fitness* secara bertahap dan fluktuatif terus meningkat mulai dari jumlah iterasi sebesar 1.000 sampai dengan 16.000. Namun rata-rata *fitness* tidak mengalami kenaikan setelah 16.000 iterasi. Sehingga, jumlah iterasi sebesar 16.000 menjadi jumlah iterasi yang optimum untuk bahan uji A13.



**Gambar 5.35 Rata-rata *fitness* untuk jumlah iterasi yang berbeda pada A14**



**Gambar 5.36 Rata-rata *fitness* untuk jumlah iterasi yang berbeda pada A15**

Hasil rata-rata *fitness* terhadap jumlah iterasi pada bahan uji A14 dapat dilihat pada Gambar 5.35. Rata-rata *fitness* secara bertahap dan fluktuatif terus meningkat mulai dari jumlah iterasi sebesar 1.000 sampai dengan 15.000. Namun rata-rata *fitness* tidak mengalami kenaikan setelah 15.000 iterasi. Sehingga, jumlah iterasi sebesar 15.000 menjadi jumlah iterasi yang optimum untuk bahan uji A14.

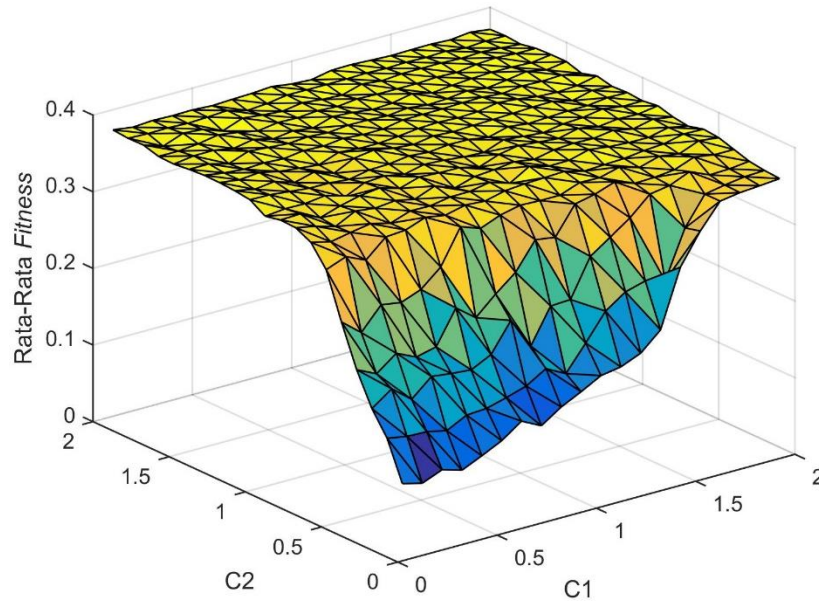
Hasil rata-rata *fitness* terhadap jumlah iterasi pada bahan uji A15 dapat dilihat pada Gambar 5.36. Rata-rata *fitness* secara bertahap dan fluktuatif terus meningkat mulai dari jumlah iterasi sebesar 1.000 sampai dengan 3.000. Namun rata-rata *fitness* tidak mengalami kenaikan setelah 3.000 iterasi. Sehingga, jumlah iterasi sebesar 3.000 menjadi jumlah iterasi yang optimum untuk bahan uji A15.

Hasil jumlah iterasi yang optimum dari masing-masing bahan uji menunjukkan hasil yang berbeda. Bahan uji A11, A12, A13, A14, dan A15 secara berturut-turut memerlukan minimal jumlah iterasi sebesar 6.000, 3.000, 16.000, 15.000, dan 3.000 untuk dapat menghasilkan solusi yang optimum. Sama seperti hasil dari skenario *swarm size* yang telah dibahas pada sub-bab 5.4, menentukan jumlah iterasi yang optimal untuk semua kemungkinan bahan uji pada optimasi formulasi pakan ayam petelur akan menjadi sangat sulit. Sehingga, strategi yang sama seperti penentuan *swarm size* yang baik digunakan dalam menentukan jumlah iterasi yang baik. Dari hasil uji yang diperoleh ditemukan bahwa jumlah iterasi yang paling tinggi terdapat pada bahan uji A13 sebesar 16.000 iterasi. Nilai parameter 16.000 iterasi menjadi jumlah iterasi yang baik karena menghasilkan solusi yang baik tanpa mengurangi rata-rata *fitness* untuk bahan uji lainnya dan dapat dijadikan acuan sebagai kriteria pemberhentian proses hibridisasi PSO yang dapat berguna untuk aplikasi praktis (Zielinski and Laur, 2007). Oleh sebab itu, 16.000 ditetapkan sebagai jumlah iterasi yang baik untuk optimasi formulasi pakan ayam petelur menggunakan hibridisasi PSO.

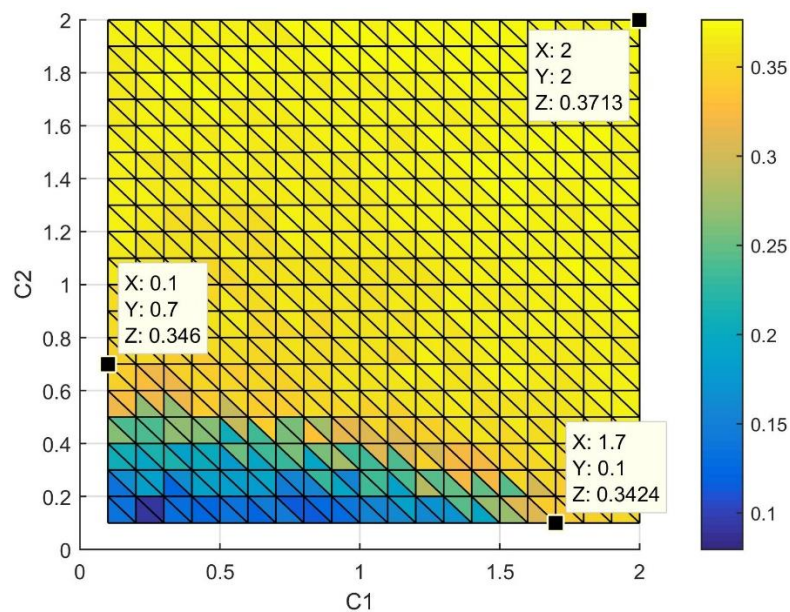
## 5.6 Parameter PSO

### 5.6.1 Koefisien Akselerasi

Hasil pengujian parameter koefisien akselerasi PSO yang meliputi koefisien akselerasi kognitif ( $C_1$ ) dan koefisien akselerasi sosial ( $C_2$ ) dapat dilihat pada Gambar 5.37 – 5.46. Gambar tersebut dibagi menjadi grafik 3 dimensi dan 2 dimensi untuk dapat melihat dampak kenaikan  $C_1$  dan  $C_2$  terhadap rata-rata *fitness*. Pengujian ini menggunakan *swarm size* dan jumlah iterasi dari hasil pengujian yang telah dibahas pada sub-bab 5.5 dan 5.6.

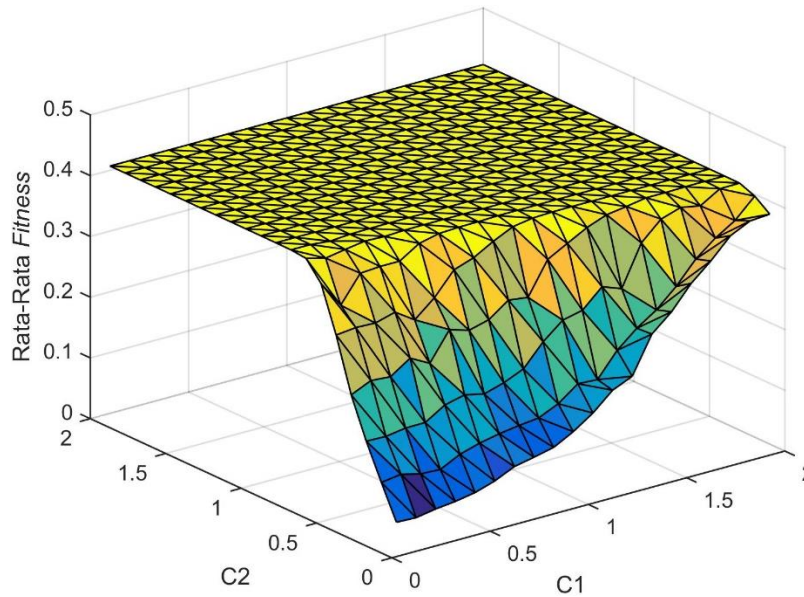


Gambar 5.37 Grafik 3D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A11



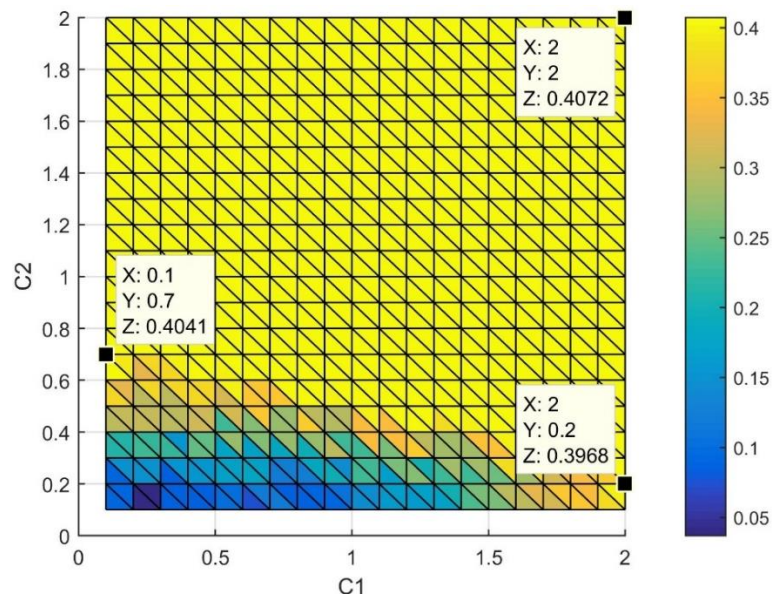
Gambar 5.38 Grafik 2D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A11



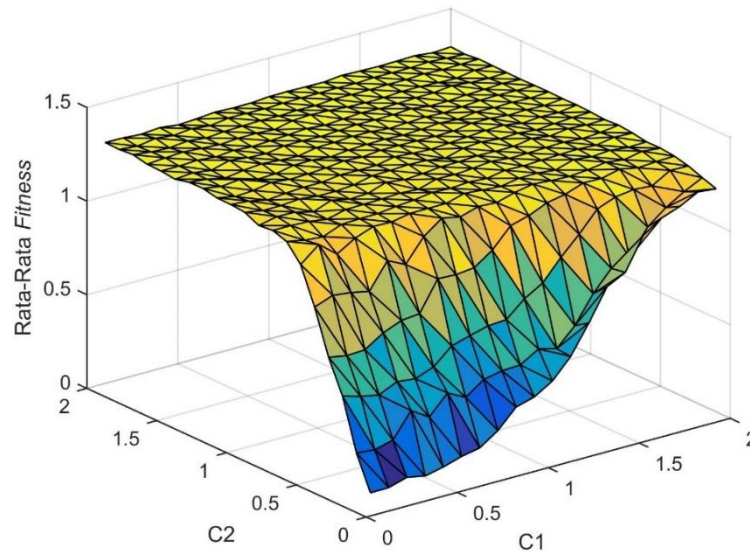


**Gambar 5.39 Grafik 3D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A12**

Grafik 2 dimensi dibuat untuk memudahkan memahami pengaruh koefisien akselerasi terhadap rata-rata *fitness* yang dihasilkan yang dapat dilihat pada Gambar 5.37, 5.39, 5.41, 5.43, dan 5.45. Nilai koefisien  $C_1$  dapat dilihat pada sumbu X, nilai koefisien  $C_2$  dapat dilihat pada sumbu Y, dan warna menunjukkan nilai rata-rata *fitness*. Pemetaan warna dengan nilai rata-rata *fitness* dapat dilihat pada *bar* yang terletak disebelah kanan dari grafik. Nilai koefisien akselerasi ditentukan berdasarkan kemampuan koefisien akselerasi menghasilkan rata-rata *fitness* yang mendekati rata-rata *fitness* maksimum yang dihasilkan. Koefisien tersebut dibandingkan dengan rekomendasi koefisien akselerasi dengan nilai  $C_1 = C_2 = 2,0$  (Yuhui Shi and Eberhart, 1998; Rezaee Jordehi and Jasni, 2013)



**Gambar 5.40 Grafik 2D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A12**



**Gambar 5.41 Grafik 3D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A14**

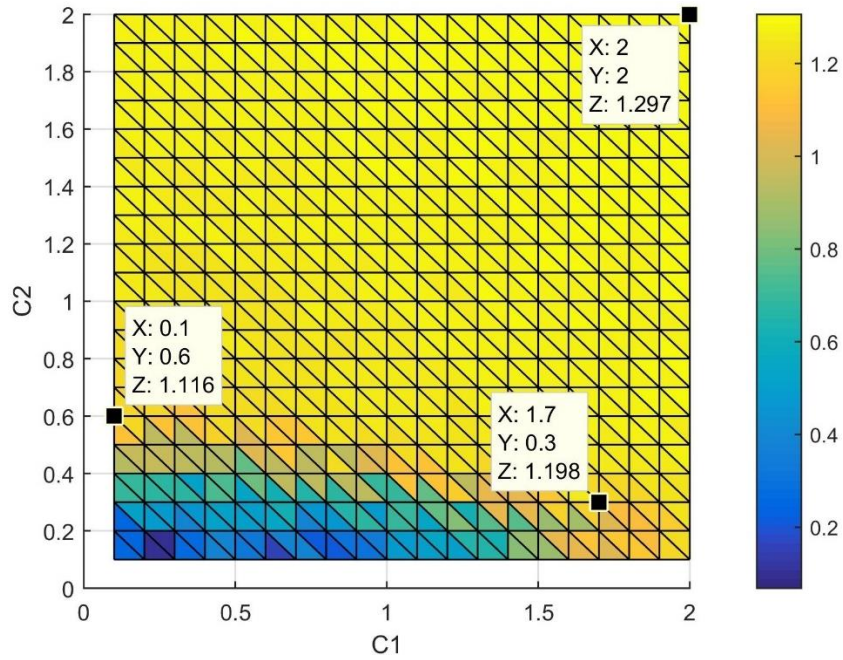
Pada Gambar 5.37 untuk formula A11 menunjukkan semakin tinggi nilai kedua koefisien semakin baik solusi yang dihasilkan yang dilihat dari warna kuning yang menunjukkan nilai rata-rata *fitness* yang mendekati nilai maksimum yang dapat dihasilkan. Pengaruh  $C_2$ , koefisien akselerasi sosial, sangat besar dibandingkan dengan  $C_1$ .

Lebih rinci seperti terlihat pada Gambar 5.38, nilai  $C_2$  yang sangat rendah tetapi dengan nilai  $C_1$  yang tinggi menghasilkan rata-rata *fitness* yang baik ( $X = 1,7$ ,  $Y = 0,1$ ). Sebaliknya dengan nilai  $C_1$  yang rendah hanya memerlukan nilai  $C_2$  yang sangat rendah pula untuk bisa menghasilkan rata-rata *fitness* yang baik ( $X = 0,1$ ,  $Y = 0,7$ ). Sedangkan, nilai  $C_1$  yang rendah dengan nilai  $C_2$  yang sangat rendah memberikan rata-rata *fitness* yang rendah pula.

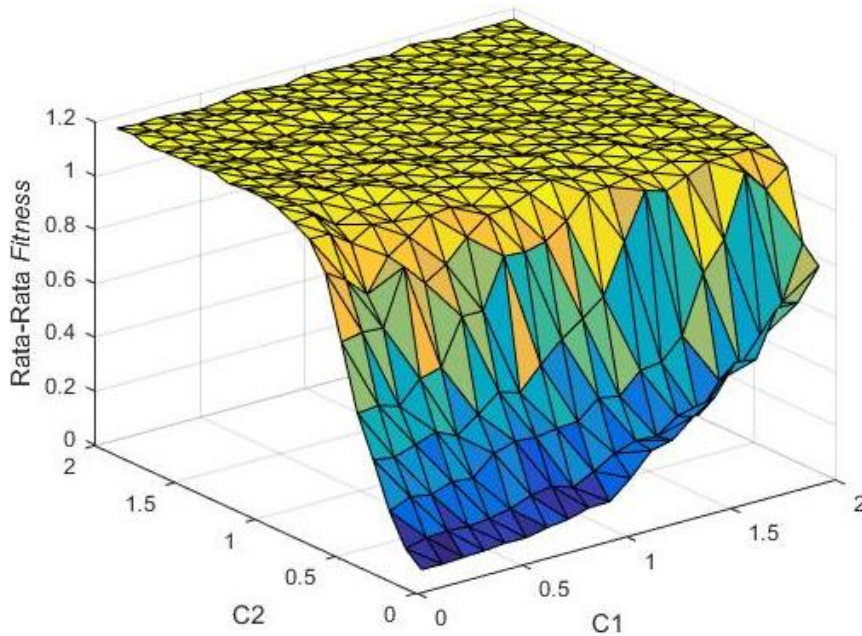
Nilai koefisien  $C_1$  diatas 1,7 dan atau nilai koefisien  $C_2$  diatas 0,7 dapat memberikan rata-rata *fitness* yang lebih baik. Jika dibandingkan dengan  $X = 2,0$  dan  $Y = 2,0$ , nilai parameter tersebut dapat menghasilkan rata-rata *fitness* yang lebih tinggi dibandingkan dengan rentang tersebut. Namun jika dilihat lebih rinci, setiap kenaikan  $C_1$  memerlukan minimal  $C_2$  yang lebih sedikit dan setiap nilai parameter  $C_1$  memerlukan minimal  $C_2$  yang berbeda-beda. Sehingga parameter koefisien yang baik untuk A11 adalah nilai  $C_1$  dan  $C_2$  yang tinggi yaitu mendekati 2,0.

Pola yang sama juga berlaku untuk formula A12, A13, A14, dan A15 yang dapat dilihat pada Gambar 5.39 – 5.46. Setiap kenaikan  $C_1$  memerlukan minimal  $C_2$  yang semakin berkurang untuk dapat memberikan rata-rata *fitness* yang tinggi. Semakin tinggi nilai parameter  $C_1$  dan  $C_2$  semakin baik rata-rata *fitness* yang dihasilkan. Namun semakin tinggi nilai tersebut semakin kecil kenaikan yang dihasilkan atau tidak terlalu signifikan atau stabil pada nilai parameter tertentu seperti pada formula A12. Seperti terlihat pada Gambar 5.39, nilai  $C_1 = 0,1$  dan  $C_2 = 0,7$  dapat menghasilkan rata-rata *fitness* sebesar 0,4041 dan nilai  $C_1 = 2,0$  dan  $C_2 = 0,2$  juga memberikan nilai yang hampir sama yaitu 0,3968. Jika nilai  $C_1$  dan  $C_2$  sama-sama

sangat tinggi dengan nilai  $C_1 = 2,0$  dan  $C_2 = 2,0$  , rata-rata *fitness* tidak dapat meningkat signifikan yaitu sebesar 0,4072.

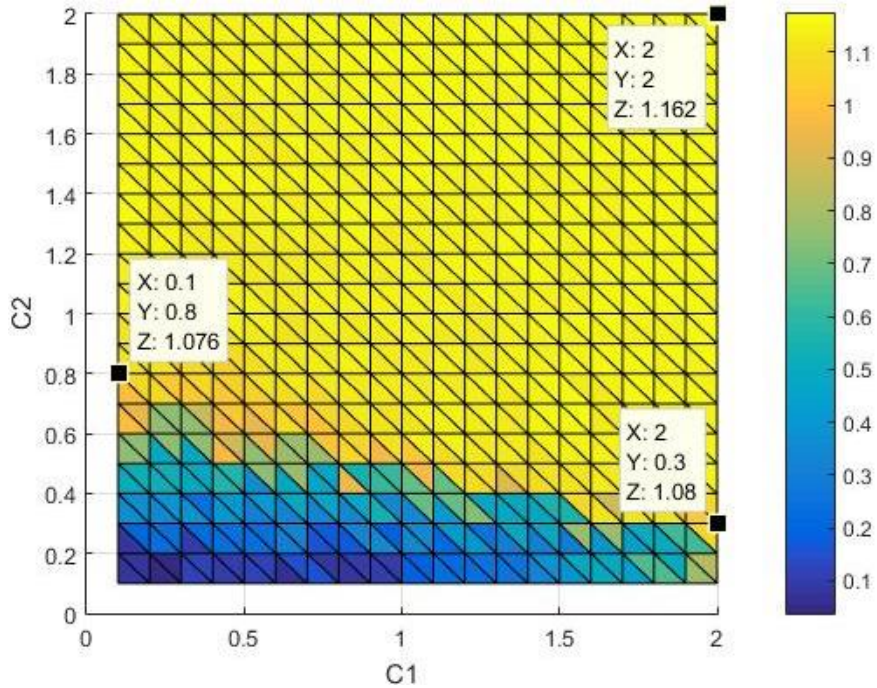


**Gambar 5.42** Grafik 2D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A14



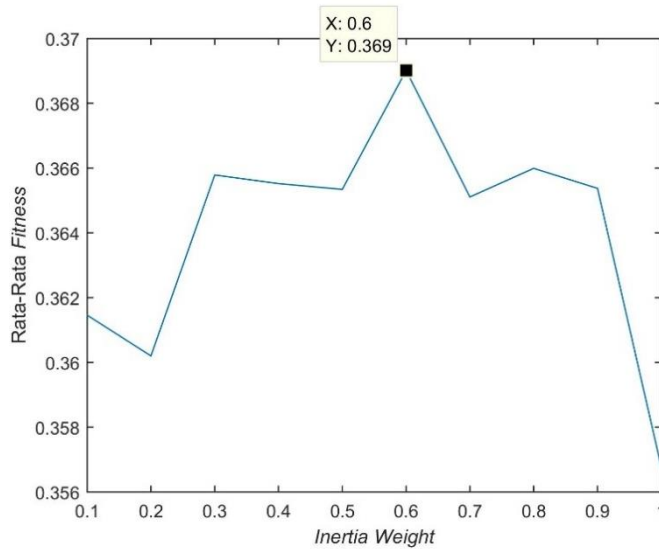
**Gambar 5.43** Grafik 3D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A15





**Gambar 5.44 Grafik 2D pengaruh  $C_1$  dan  $C_2$  terhadap rata-rata *fitness* pada A15**

Dari semua grafik pengaruh koefisien akselerasi terhadap rata-rata *fitness* menunjukkan rata-rata *fitness* yang baik dapat dihasilkan jika melibatkan parameter koefisien akselerasi dengan nilai yang tinggi. Sehingga nilai  $C_1 = 2,0$  dan  $C_2 = 2,0$  ditetapkan sebagai parameter yang baik yang akan digunakan pada pengujian selanjutnya yang sejalan dengan rekomendasi parameter koefisien akselerasi pada studi yang lain (Yuhui Shi and Eberhart, 1998; Rezaee Jordehi and Jasni, 2013).

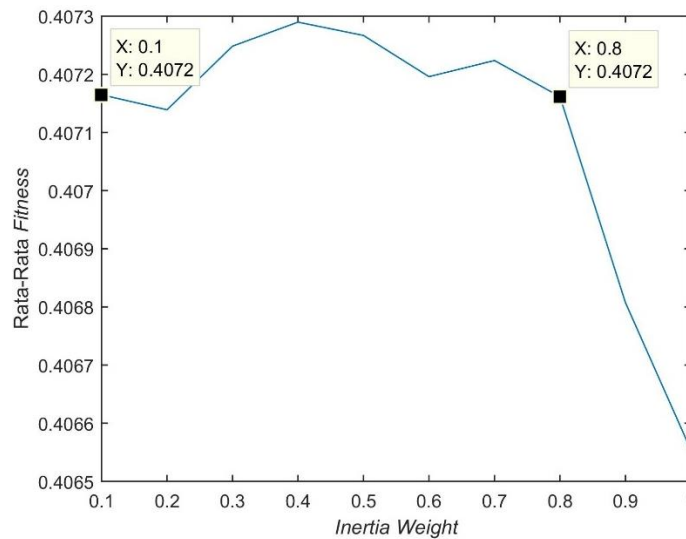


**Gambar 5.45 Rata-rata *fitness* untuk *inertia weight* yang berbeda pada A11**

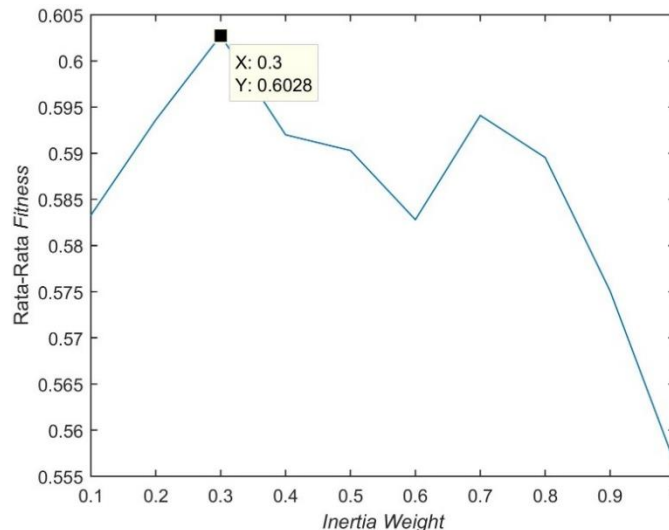
### 5.6.2 Inertia Weight

Hasil pengujian *inertia weight* ( $w$ ) dengan penyesuaian nilai antara 0 sampai 1 dapat dilihat pada Gambar 5.47 – 5.51. Pengujian ini menggunakan *swarm size*, jumlah iterasi, dan koefisien akselerasi dari hasil pengujian yang telah dibahas pada sub-bab 5.5, 5.6 dan 5.7.1.

Pada formula A11 seperti terlihat pada Gambar 5.47, rata-rata *fitness* meningkat fluktuatif dari 0,1 sampai 0,6 lalu menurun secara bertahap antara 0,7 sampai 0,9 dan menurun drastis pada  $w = 1,0$ . Pada kasus formula A11, 0,6 merupakan nilai parameter  $w$  yang terbaik.



Gambar 5.46 Rata-rata *fitness* untuk *inertia weight* yang berbeda pada A12

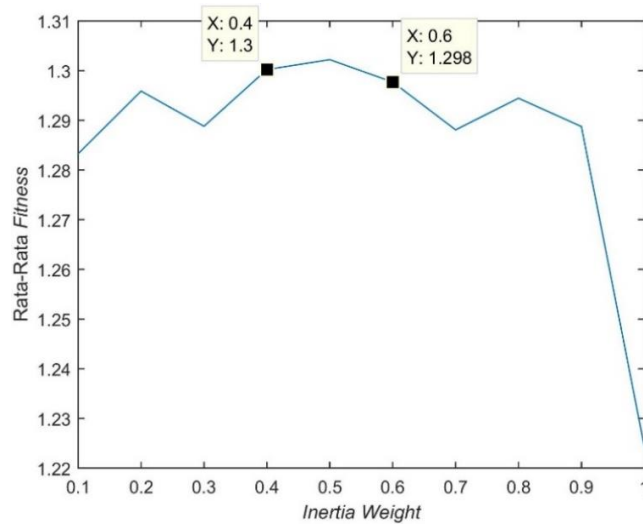


Gambar 5.47 Rata-rata *fitness* untuk *inertia weight* yang berbeda pada A13

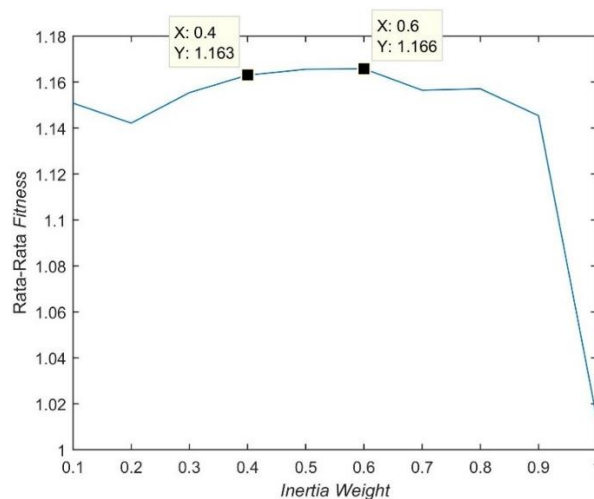
Pada formula A12 seperti terlihat pada Gambar 5.48,  $w = 0,4$  menghasilkan rata-rata *fitness* tertinggi, namun jika dibandingkan dengan 0,1 ataupun 0,8 peningkatan tersebut tidak terlalu signifikan jika dilihat dari nilai rata-rata *fitness*. Sehingga dikarenakan parameter *inertia weight* antara 0,1 sampai 0,8 tidak

menghasilkan peningkatan yang signifikan, maka nilai dalam rentang tersebut merupakan nilai *inertia weight* yang baik untuk dipilih pada kasus formula A12.

Pada formula A13 seperti terlihat pada Gambar 5.49, rata-rata *fitness* meningkat secara bertahap dari 0,1 sampai 0,3 dan menurun diatas 0,3. Pada kasus formula A13,  $w = 0,3$  merupakan parameter yang terbaik. Pada formula A14 seperti terlihat pada Gambar 5.50, rata-rata *fitness* meningkat fluktuatif mulai dari 0,1 sampai 0,4 dan tidak ada peningkatan yang signifikan nilai  $w$  antara 0,4 sampai 0,6. Diatas 0,6, rata-rata *fitness* terjadi penurunan seperti yang ditemui pada formula lainnya. Sehingga pada kasus A14, nilai  $w$  antara 0,4 dan 0,6 merupakan parameter yang terbaik dalam menghasilkan rata-rata *fitness* yang tinggi. Hasil parameter yang terbaik yang sama juga ditemukan untuk formula A15 seperti terlihat pada Gambar 5.51 yaitu sebesar 0,4 sampai 0,6.



**Gambar 5.48 Rata-rata *fitness* untuk *inertia weight* yang berbeda pada A14**



**Gambar 5.49 Rata-rata *fitness* untuk *inertia weight* yang berbeda pada A15**

Dara hasil nilai parameter  $w$  yang terbaik diketahui bahwa untuk A11 nilai parameter  $w$  yang terbaik adalah 0,6. Untuk A12, nilai parameter  $w$  yang terbaik

adalah antara 0,1 sampai 0,8. Untuk formula A13, nilai parameter  $w$  sebesar 0,3 dapat memberikan rata-rata *fitness* tertinggi. Sedangkan untuk formula A14 dan A15, nilai parameter  $w$  antara 0,4 dan 0,6 merupakan parameter terbaik.

Dari hasil tiap-tiap parameter  $w$  untuk tiap-tiap formula menunjukkan bahwa setiap pilihan bahan pakan yang akan di formulasikan membutuhkan nilai parameter  $w$  yang berbeda. Selain itu nilai  $w$  yang tinggi atau mendekati 1,0 akan menurunkan rata-rata *fitness* yang menunjukkan rendahnya kualitas solusi hasil formulasi. Semua formula menunjukkan nilai  $w = 1,0$  merupakan pilihan yang buruk karena akan menurunkan rata-rata *fitness* sangat signifikan. Selain itu pemilihan parameter  $w$  yang mendekati 0 menunjukkan hasil formulasi yang belum optimal yang ditunjukkan oleh rata-rata *fitness* yang masih bisa ditingkatkan ketika nilai  $w$  dinaikkan. Sehingga pemilihan parameter  $w$  yang baik sebaiknya tidak terlalu kecil dan tidak terlalu tinggi walaupun tidak ada satu nilai parameter  $w$  yang terbaik untuk seluruh formula.

Dari ketiga formula, A12, A14, dan A15, pilihan parameter  $w$  antara 0.4 dan 0.6 menjadi pilihan terbaik yang diperoleh melalui interseksi dari hasil parameter  $w$  untuk ketiga formula tersebut. Ketika mempertimbangkan semua sampel formula yang melibatkan formula A11 dan A13, tidak ada parameter terbaik untuk semua formula. Namun jika 4 formula dipertimbangkan tanpa melibatkan A13, parameter yang terbaik menjadi 0,6. Tidak ada parameter terbaik untuk semua formula sehingga berdasarkan 4 formula ditemukan bahwa parameter  $w$  sebesar 0,6 menjadi parameter yang baik untuk dipilih mengingat parameter tersebut tidak terlalu kecil dan tidak terlalu besar dimana dapat mengurangi kualitas solusi formulasi.

## 5.7 Komparasi

Sebelum uji komparasi dilakukan, parameter GA perlu dilakukan *tuning* untuk didapatkan solusi yang optimum. Dengan parameter tersebut diharapkan dapat bersanding dengan MSPSO dan algoritme komparasi yang lain yang telah dilakukan penyesuaian sebelumnya. Oleh sebab itu, parameter GA dilakukan *tuning* yang dibahas pada sub-bab 5.6.1 dan hasil komparasi berupa rata-rata *fitness*, penalti, biaya, waktu komputasi dan standar deviasi dibahas lebih rinci pada sub-bab 5.6.2

### 5.7.1 Parameter *Crossover Rate* dan *Mutation Rate* pada GA

Agar pengujian komparasi terhadap GA adil dengan komparasi lainnya, parameter GA disesuaikan terlebih dahulu untuk dapat menemukan parameter yang terbaik agar dapat disandingkan dengan MSPSO dan PSO. Hasil uji parameter *crossover rate* ( $cr$ ) dan *mutation rate* ( $mr$ ) dapat dilihat pada Tabel 5.6.

Nilai masing-masing parameter ditetapkan berjumlah 1,0 dengan kenaikan/penurunan tiap-tiap parameter sebesar 0,1 untuk formula A11. Dari hasil pengujian tersebut menunjukkan  $cr$  dengan nilai 0,7 dan  $mr$  dengan nilai 0,3 dapat memberikan nilai *fitness* terbaik untuk GA dibandingkan dengan hasil kombinasi nilai parameter yang lain. Oleh sebab itu, ditetapkan  $cr$  sebesar 0,7 dan

*mr* sebesar 0,3 sebagai nilai parameter GA yang baik untuk permasalahan optimasi pakan ayam petelur. Parameter tersebut akan digunakan oleh GA untuk dikomparasikan dengan algoritme yang lain.

**Tabel 5.4 Hasil pengujian parameter *cr* dan *mr***

<i>cr</i>	<i>mr</i>	Rata-rata <i>fitness</i>
0	1	0,282246
0,1	0,9	0,276853
0,2	0,8	0,25441
0,3	0,7	0,269226
0,4	0,6	0,303298
0,5	0,5	0,294279
0,6	0,4	0,292369
<b>0,7</b>	<b>0,3</b>	<b>0,304387</b>
0,8	0,2	0,289948
0,9	0,1	0,298292
1	0,0	0,278444

## 5.7.2 Komparasi Algoritme

Komparasi beberapa algoritme yang telah diterapkan untuk optimasi pakan ternak dijabarkan dalam beberapa bahasan meliputi *fitness* untuk melihat algoritme yang dapat memberikan rata-rata *fitness* tertinggi. Lalu penalti dan biaya, untuk melihat kemampuan algoritme komparasi dalam menghasilkan penalti dan biaya yang rendah. Lalu waktu komputasi, untuk melihat seberapa cepat algoritme komparasi dalam menghasilkan solusi yang optimum. Dan standar deviasi untuk melihat seberapa stabil algoritme komparasi dari formula yang fluktuatif yang dihasilkan.

### 5.7.2.1 *Fitness*

Hasil skenario pengujian komparasi untuk Algoritme Genetika (GA), Hibridisasi Algoritme Genetika Adaptif dengan *Simulated Annealing* (AGASA), Particle Swarm Optimization (PSO), dan Hibridisasi PSO yang merupakan *Multi Swarm Particle Swarm Optimization* (MSPSO) dengan 5 bahan uji yang berbeda (B11 – B15) dapat dilihat pada Tabel 5.7. Hasil uji dibulatkan kedalam ketelitian 5 angka dibelakang koma. Semakin tinggi nilai *fitness* menunjukkan semakin baik solusi yang dihasilkan yang menunjukkan rendahnya penalti dan/atau rendahnya biaya pakan.

Rata-rata *fitness* yang dihasilkan oleh algoritme evolusi seperti GA dan AGASA lebih rendah dibandingkan dengan PSO dan MSPSO. Semua formula yang dihasilkan oleh GA menunjukkan rata-rata *fitness* paling rendah dibandingkan dengan algoritme komparasi lainnya. Hibridisasi GA adaptif dengan *simulated annealing* dapat meningkatkan rata-rata *fitness* jika dibandingkan dengan GA. Namun pendekatan algoritme evolusi tersebut masih belum mampu

menghasilkan rata-rata *fitness* yang lebih tinggi dibandingkan dengan pendekatan *swarm intelligence* khususnya PSO.

**Tabel 5.5 Hasil uji rata-rata *fitness* untuk semua algoritme komparasi**

Formula	Rata-Rata <i>Fitness</i>			
	GA	AGASA	PSO	MSPSO
B11	0,751287480	0,789272155	0,888860280	<b>0,921979767</b>
B12	0,132474527	0,162040519	0,176900793	<b>0,183160686</b>
B13	1,084670584	1,291235804	1,363657097	<b>1,408007168</b>
B14	0,334879716	0,393736283	0,475759291	<b>0,513210327</b>
B15	1,619706181	1,895306038	2,039038192	<b>2,075200965</b>
Rata-Rata	0,784603697	0,906318160	0,988843130	<b>1,020311783</b>

Seperti terlihat pada Tabel 5.7, MSPSO dapat menghasilkan rata-rata *fitness* tertinggi dibandingkan dengan semua algoritme komparasi lain yang ditunjukkan dengan huruf cetak tebal. Data hasil uji tersebut menunjukkan bahwa MSPSO merupakan algoritme yang terbaik dalam hal nilai *fitness* jika dibandingkan dengan GA, AGASA, dan PSO standar.

Membagi *swarm* menjadi beberapa *sub-swarm* yang memanfaatkan peluang pergerakan biseksi dan fase pergerakan yaitu migrasi dan agregasi dapat membuat partikel menemukan solusi yang lebih baik. Dengan memanfaatkan *sub-swarm* akan meningkatkan keberagaman partikel dan memungkinkan memiliki arah pencarian yang berbeda sehingga meningkatkan eksplorasi partikel dan cenderung menghasilkan solusi yang lebih baik. Ketika informasi penting terdapat pada partikel lain, maka suatu partikel dapat bergerak menuju partikel baik yang lain dengan memanfaatkan pergerakan biseksi yang merepresentasikan komunikasi antar partikel yang tidak dimiliki oleh PSO standar. Selain itu dengan membagi pergerakan partikel menjadi dua fase akan membuat semua informasi yang disimpan oleh partikel lebih dimanfaatkan (karena adanya fase agregasi yang menggabungkan semua *sub-swarm* menjadi satu *swarm* kembali) sehingga eksploitasi terhadap ruang pencarian akan meningkat dan solusi yang lebih baik dapat dihasilkan.

### 5.7.2.2 Penalti dan Biaya

Hasil skenario pengujian komparasi untuk rata-rata penalti dan biaya dapat dilihat pada Tabel 5.8 dimana nilai terendah ditunjukkan dengan huruf cetak tebal. Total penalti diperoleh dari penjumlahan antara penalti non-asam amino, penalti asam amino, dan penalti batasan komposisi. Semakin rendah total penalti menunjukkan bahwa nutrisi yang terkandung dalam pakan dapat memenuhi semua kendala kebutuhan nutrisi, seimbangannya asam amino, dan kendala maksimum komposisi pakan ayam petelur.

Semua formula yang dihasilkan oleh MSPSO dapat memberikan rata-rata total penalti paling rendah dibandingkan dengan GA, AGASA, dan PSO. Sedangkan rata-

rata penalti paling tinggi dihasilkan oleh formulasi GA. AGASA dapat menghasilkan formula dengan penalti yang lebih rendah dibandingkan dengan GA, namun tidak lebih baik jika dibandingkan dengan PSO dan MSPSO. Pada kasus formulasi ayam petelur, PSO lebih unggul dalam hal penalti jika dibandingkan dengan pendekatan algoritme evolusi seperti GA walaupun GA ditingkatkan dengan teknik adaptif dan hibridisasi dengan *simulated annealing*. Dari hasil rata-rata total penalti tersebut MSPSO terbukti mampu menghasilkan penalti yang lebih rendah dibandingkan dengan algoritme komparasi lainnya.

Dalam hal rendahnya biaya, MSPSO tidak dapat memberikan solusi biaya yang paling minimum untuk semua formula. Hal ini sangat dimungkinkan karena harga dari bahan pakan yang meningkat sebanding dengan penurunan penalti. Namun, pada kasus formula B15, MSPSO dapat memberikan harga paling minimum dengan penalti terendah. Hal tersebut menunjukkan bahwa dalam kasus pilhan bahan pakan tertentu MSPSO dapat menghasilkan solusi formula dengan penalti yang rendah namun juga dengan biaya paling rendah.

**Tabel 5.6 Hasil uji rata-rata penalti dan biaya untuk semua algoritme komparasi**

Formula	Algoritme	Rata-Rata Total Penalti	Rata-Rata Penalti Non-Asam Amino	Rata-Rata Penalti Asam Amino	Rata-Rata Penalti Batasan Komposisi	Rata-Rata Biaya (Rp. /100 Kg.)
B11	GA	1,05721	0,84853	0,20868	0,0000E+00	<b>723.820,92</b>
	AGASA	0,96842	0,74059	0,22783	0,0000E+00	762.739,17
	PSO	0,82348	0,57423	0,24925	2,4037E-06	764.838,63
	MSPSO	<b>0,77646</b>	0,51717	0,25928	2,0932E-08	776.194,53
B12	GA	7,56963	4,71861	0,07028	2,7807E+00	476.833,77
	AGASA	6,17411	4,59586	0,05827	1,5200E+00	471.159,79
	PSO	5,64538	4,58870	0,05250	1,0042E+00	<b>466.113,50</b>
	MSPSO	<b>5,44493</b>	4,42589	0,05247	9,6657E-01	469.386,54
B13	GA	0,91911	0,60689	0,31222	3,8733E-07	<b>812.973,16</b>
	AGASA	0,73407	0,44582	0,28825	4,2342E-07	833.099,39
	PSO	0,68602	0,38041	0,30561	0,0000E+00	933.045,68
	MSPSO	<b>0,65760</b>	0,35082	0,30678	0,0000E+00	1.008.076,51
B14	GA	2,61677	2,53485	0,08193	0,0000E+00	<b>834.797,64</b>
	AGASA	2,13732	2,06301	0,07431	0,0000E+00	891.588,68
	PSO	1,64747	1,56606	0,08141	0,0000E+00	974.179,20
	MSPSO	<b>1,45902</b>	1,36302	0,09600	5,8353E-06	1.036.036,85
B15	GA	0,56464	0,44351	0,12114	0,0000E+00	1.023.818,55
	AGASA	0,48070	0,35305	0,12765	0,0000E+00	925.015,24
	PSO	0,44725	0,35557	0,09168	0,0000E+00	869.017,47
	MSPSO	<b>0,43921</b>	0,34757	0,09164	0,0000E+00	<b>861.502,65</b>

**Tabel 5.7 Hasil uji rata-rata penalti dan biaya untuk semua algoritme komparasi dengan Standar Nasional Indonesia (SNI)**

Formula	Algoritme	Rata-Rata Total Penalty	Rata-Rata Penalty Non-Asam Amino	Rata-Rata Penalty Asam Amino	Rata-Rata Penalty Batasan Komposisi	Rata-Rata Biaya (Rp. /100 Kg.)
B11	GA	2,04779E-03	0,00205	1,15173E-06	0	452.178,19
	AGASA	3,37688E-03	0,00336	1,55719E-05	0	449.935,37
	PSO	2,34955E-04	0,00021	2,06954E-05	0	447.670,83
	MSPSO	<b>8,23374E-05</b>	0,00006	2,08032E-05	0	<b>446.430,34</b>
B12	GA	1,39943E-06	0	1,39943E-06	0	442.735,08
	AGASA	8,21166E-07	0	8,21166E-07	0	439.159,57
	PSO	<b>1,96680E-08</b>	0	1,96680E-08	0	435.147,74
	MSPSO	8,12822E-06	0	8,12822E-06	0	<b>428.325,66</b>
B13	GA	0,01449	0,01011	0,00437	0	587.511,36
	AGASA	0,01029	0,00896	0,00133	0	523.605,13
	PSO	0,00980	0,00870	0,00110	0	476.994,58
	MSPSO	<b>0,00835</b>	0,00675	0,00160	0	<b>460.763,82</b>
B14	GA	3,27994E-05	0	3,27994E-05	0	447.160,17
	AGASA	2,10351E-05	0	2,10351E-05	0	454.443,84
	PSO	<b>4,94345E-07</b>	0	4,94345E-07	0	444.413,90
	MSPSO	3,19159E-06	0	3,19159E-06	0	<b>438.961,14</b>
B15	GA	<b>1,70500E-07</b>	0	1,70500E-07	0	1.047.814,95
	AGASA	2,78955E-06	0	2,78955E-06	0	852.934,47
	PSO	1,29322E-05	0	1,29322E-05	0	718.820,81
	MSPSO	4,72593E-05	0	4,72593E-05	0	<b>684.439,42</b>

Hasil pengujian tersebut menunjukkan adanya penalti yang tidak mendekati nol kecuali untuk batasan komposisi, sehingga pendekatan komputasi evolusioner akan menyeimbangkan antara penalti non-asam amino dan asam amino dengan biaya melalui fungsi *fitness* yang telah didefinisikan. Untuk membuktikan bahwa model hibridisasi PSO dapat menghemat biaya formulasi pakan, maka perlu mengurangi kompleksitas ruang pencarian yang tidak terlalu kompleks seperti sebelumnya. Salah satu yang mempengaruhi kompleksitas ruang pencarian pada formulasi pakan ternak adalah kebutuhan nutrisi yang dijadikan sebagai kendala pada penerapan komputasi evolusioner seperti hibridisasi PSO. Oleh sebab itu, kebutuhan nutrisi sesuai Standar Nasional Indonesia (SNI) digunakan karena kebutuhan zat makanan pada SNI tidak terpaku pada batasan jangkauan saja tetapi memiliki batasan minimum dan maksimum. Sehingga solusi optimum cenderung lebih mudah ditemukan. Secara rinci kebutuhan zat makanan atau nutrisi ayam petelur berdasarkan SNI dapat dilihat pada Lampiran D.

Hasil uji komparasi rata-rata penalti dan biaya berdasarkan SNI dapat dilihat pada Tabel 5.9. Nilai cetak tebal menunjukkan rata-rata penalti dan biaya paling



rendah yang sekaligus menunjukkan efektifitas algoritme tersebut. Seperti terlihat pada Tabel 5.9, semua algoritme komparasi dapat memberikan penalti yang mendekati nol yang menunjukkan semua kendala dapat dipenuhi. Namun, rata-rata biaya terendah dihasilkan oleh MSPSO. Hasil uji tersebut menunjukkan bahwa jika kompleksitas ruang pencarian menyebabkan algoritme tersebut dapat memberikan hasil penalti yang mendekati nol, maka MSPSO mampu memberikan rata-rata biaya terendah dibandingkan dengan algoritme komparasi yang lain. Ini membuktikan bahwa MSPSO merupakan algoritme yang efektif dalam menemukan solusi formula yang optimum dengan biaya terendah.

### 5.7.2.3 Waktu Komputasi

Hasil skenario pengujian komparasi untuk waktu komputasi dapat dilihat pada Tabel 5.10 dimana waktu komputasi terendah ditunjukkan dengan huruf cetak tebal. Semakin kecil waktu komputasi menunjukkan semakin cepat algoritme tersebut dalam menemukan solusi yang optimum sehingga solusi praktis dapat diperoleh.

**Tabel 5.8 Hasil uji rata-rata waktu komputasi untuk semua algoritme komparasi**

Formula	Rata-Rata Waktu (detik)			
	GA	AGASA	PSO	MSPSO
B11	81,504	278,550	37,366	<b>31,370</b>
B12	94,299	382,553	33,229	<b>32,279</b>
B13	89,735	275,937	45,228	<b>36,196</b>
B14	104,861	386,479	43,594	<b>37,179</b>
B15	106,611	328,841	54,436	<b>46,640</b>

Seperti terlihat pada Tabel 5.10, AGASA memerlukan waktu yang paling lama dibandingkan algoritme komparasi lainnya. Penambahan pencarian solusi dengan menggunakan *simulated annealing* akan meningkatkan kompleksitas waktu komputasi. Terlebih, rata-rata *fitness* AGASA lebih kecil dibandingkan PSO standar untuk semua bahan uji. Hal ini menunjukkan AGASA kurang efisien dalam menemukan solusi yang optimum pada permasalahan optimasi pakan ternak khususnya ayam petelur.

Sebaliknya, rata-rata waktu komputasi terendah ditemukan pada MSPSO. Jika melihat runtutan langkah pergerakan PSO, MSPSO harusnya memberikan waktu komputasi yang lebih lama dibandingkan dengan PSO. Karena MSPSO melibatkan migrasi setiap periode tertentu dan adanya proses penggabungan semua *sub-swarm* menjadi satu *swarm* pada fase agregasi. MSPSO memerlukan waktu komputasi yang lebih cepat dimungkinkan karena adanya *multi-core* pada CPU yang ditangani langsung oleh sistem operasi dimana masing-masing *sub-swarm* di proses pada CPU yang berbeda secara bersamaan. Sehingga dimungkinkan karena hal tersebut MSPSO memerlukan waktu yang lebih sedikit dibandingkan dengan PSO. Jika hal tersebut terbukti benar, maka pendekatan *multi-swarm* pada PSO

akan memberikan manfaat yang besar jika dipadukan dengan komputasi paralel dimana MSPSO dapat memberikan solusi yang lebih baik dan lebih cepat dibandingkan dengan PSO standar. Namun hal tersebut harus dibuktikan lebih lanjut secara empiris ataupun teoritis bagaimana MSPSO memerlukan waktu komputasi yang lebih kecil dibandingkan dengan PSO Standar.

#### 5.7.2.4 Standar Deviasi

Hasil skenario pengujian komparasi untuk standar deviasi dapat dilihat pada Tabel 5.11 dimana huruf cetak tebal menunjukkan standar deviasi terkecil. Karena semua algoritme komparasi termasuk dalam kelas komputasi evolusioner yang merupakan optimasi stokastik dimana solusi yang dihasilkan memungkinkan berbeda-beda setiap kali dijalankan, maka standar deviasi menjadi peting untuk dianalisa. Semakin kecil standar deviasi menunjukkan semakin stabil solusi yang dihasilkan atau solusi yang dihasilkan tidak jauh berbeda jika algoritme tersebut dijalankan kembali.

**Tabel 5.9 Hasil uji rata-rata standar deviasi untuk semua algoritme komparasi**

Formula	Rata-Rata Standar Deviasi			
	GA	AGASA	PSO	MSPSO
B11	0,048095732	0,036006845	0,021669053	<b>0,018873157</b>
B12	0,009966503	0,008988457	0,006523191	<b>0,000873764</b>
B13	0,196623889	0,032971752	0,029021469	<b>0,013304586</b>
B14	0,02440654	0,027176501	0,018277233	<b>0,008324574</b>
B15	0,067183806	0,032394432	<b>0,028751302</b>	0,029253444

Dari hasil rata-rata standar deviasi yang terlihat pada Tabel 5.11. 4 dari 5 bahan uji (kecuali B14), GA memberikan rata-rata standar deviasi paling tinggi dibandingkan algoritme komparasi yang lain. Ini menunjukkan GA menghasilkan solusi yang fluktuatif dengan jangkauan yang cukup tinggi. Sebaliknya, MSPSO memberikan rata-rata standar deviasi paling kecil untuk hampir semua bahan uji (kecuali B15). Hasil tersebut menunjukkan kestabilan algoritme MSPSO dalam menghasilkan formula yang optimum.

#### 5.7.2.5 Penambahan Beban pada *Soft Constraint*

Selain menguji tingkat optimalitas, waktu komputasi, dan kestabilan solusi yang dihasilkan oleh MSPSO, MSPSO juga diuji dalam situasi yang berbeda seperti yang telah dibahas sebelumnya dengan menambahkan beban lebih berat pada salah satu kendala. Hal ini ditujukan untuk mengukur seberapa tangguh model hibridisasi tersebut jika diterapkan pada kasus yang lain atau penambahan kendala lainnya.

**Tabel 5.10 Daftar bahan pakan uji untuk perubahan fungsi *fitness***

No.	Bahan Pakan Uji	Bahan Pakan
1	C8	0,3,9,10,18,21,26,28
2	C9	0,5,6,11,18,21,25,26,29
3	C10	0,1,3,8,10,15,18,21,26,27
4	C11	0,1,3,4,6,10,12,14,16,21,26
5	C12	0,1,2,3,6,9,18,19,21,22,26,28

$$pMI(\bar{X}) = \sum_{i=1}^{|H|} \begin{cases} (x_i - \max(h_i, f))^2, & \max(h_i, f) < x_i \\ 0, & \max(h_i, f) \geq x_i \end{cases} \quad (5.1)$$

**Tabel 5.11 Perbandingan rata-rata *fitness* dan penalti batasan komposisi antara PSO dan MSPSO sebelum penambahan beban pada kendala batasan komposisi**

Formula	Algoritme	Rata-Rata <i>Fitness</i>	Rata-Rata Penalti Batasan Komposisi
C8	PSO	0.070987837	4.701084223
	MSPSO	<b>0.073706215</b>	6.074402738
C9	PSO	<b>0.037409222</b>	15.11255673
	MSPSO	0.037405294	14.97874553
C10	PSO	<b>0.180513537</b>	2.11114461
	MSPSO	<b>0.180513537</b>	2.11114461
C11	PSO	0.108682618	4.169174061
	MSPSO	<b>0.111775395</b>	4.085789226
C12	PSO	0.061855882	4.624128299
	MSPSO	<b>0.062255751</b>	3.02286794

Dari hasil uji pada bahasan penalti dan biaya sebagaimana dibahas pada sub-bab 5.8.2.2, semua formula menghasilkan penalti batasan komposisi mendekati nol yang menunjukkan bahwa formula yang dihasilkan tidak cepat membuat ayam petelur kenyang yang dapat dikatakan bahwa asupan yang diperoleh lebih baik. Tetapi bagaimana jika suatu pilihan bahan pakan oleh peternak menghasilkan formula dengan penalti batasan komposisi yang cukup tinggi ? Sehingga bahan pakan dipilih berdasarkan tingkat penalti batasan komposisi yang cukup tinggi yang berpotensi untuk diturunkan dengan menambah beban pada kendala batasan komposisi. Pilihan bahan pakan tersebut didefinisikan pada Tabel 5.12. Sedangkan penalti batasan komposisi didefinisikan pada Persamaan 5.1.

Sebelum beban pada kendala batasan komposisi ditambah dengan menggunakan Persamaan 5.1, PSO dan MSPSO dijalankan terlebih dahulu menggunakan bahan pakan pada Tabel 5.12. Hasil rata-rata *fitness* dan rata-rata penalti batasan komposisi dapat dilihat pada Tabel 5.13. Hampir semua hasil uji menunjukkan bahwa MSPSO dapat memberikan rata-rata *fitness* yang lebih tinggi

atau sama pada formula C8, C10, C11, dan C12. Pada formula C9, PSO memberikan rata-rata *fitness* lebih tinggi dibandingkan dengan MSPSO namun peningkatan tidak signifikan. Semua algoritme menunjukkan rata-rata penalti batasan komposisi diatas 2% dan ada yang mencapai 15% yang terdapat pada formula C9.

**Tabel 5.12 Perbandingan rata-rata *fitness* dan penalti batasan komposisi antara PSO dan MSPSO selesai penambahan beban pada kendala batasan komposisi**

Formula	Algoritme	Rata-Rata <i>Fitness</i>	Rata-Rata Penalti Batasan Komposisi
C8	PSO	0.066056662	1.051070803
	MSPSO	<b>0.06607081</b>	1.102210118
C9	PSO	<b>0.030641265</b>	4.288209742
	MSPSO	<b>0.030641265</b>	4.288280505
C10	PSO	0.163347053	1.222344319
	MSPSO	<b>0.163425436</b>	1.193714872
C11	PSO	0.078312453	1.144530008
	MSPSO	<b>0.079140931</b>	1.147228796
C12	PSO	0.064411848	0.258129838
	MSPSO	<b>0.064475992</b>	0.232481128

Setelah beban pada pemberian penalti untuk kendala batasan komposisi ditambah dengan menggunakan Persamaan 5.1, MSPSO tetap dapat memberikan rata-rata *fitness* tertinggi atau sama seperti terlihat pada Tabel 5.13. Namun peningkatan yang dapat dihasilkan oleh MSPSO tidak terlalu signifikan. Dampak dari penambahan beban dapat menurunkan rata-rata penalti batasan komposisi untuk semua formula. Dampak yang signifikan diperoleh dari formula C9 yang semula memiliki penalti sebesar 14,979 dari MSPSO menjadi 4,288. Ini menunjukkan bahwa walaupun ada perubahan kendala yang menyesuaikan pada kasus pakan ternak tertentu MSPSO mampu memberikan rata-rata *fitness* yang lebih baik.

Hasil pengujian tersebut juga menunjukkan bahwa MSPSO akan sangat efektif untuk digunakan jika kombinasi pilihan bahan pakan dapat menghasilkan penalti batasan komposisi mendekati nol seperti terlihat pada bahasan sub-bab 5.8.2.1 atau total penalti mendekati nol seperti terlihat pada bahasan sub-bab 5.8.2.2 dengan menggunakan standar SNI. Karena peningkatan yang diberikan tidak terlalu signifikan jika didapati total penalti yang cukup tinggi yang menunjukkan bahwa ruang pencarian dimungkinkan tidak memiliki akar solusi yang optimum.

## 5.8 Hasil Formulasi

Sub-bab ini membahas hasil formulasi yang dihasilkan oleh hibridisasi PSO menurut pakar pakan unggas yang disimulasikan pada fase *layer*. Bahan pakan yang dipilih sangat tergantung dari ketersediaan bahan pakan yang ada disuatu daerah. Pada simulasi ini, bahan pakan dipilih menurut ketersediaan bahan pakan di lapang menurut pakar. Bahan pakan yang digunakan antara lain, dedak halus, jagung kuning, bungkil kedele, bungkil kelapa, tepung ikan (Menhaden), tepung

daging, tepung tulang, dan minyak kelapa (kadungan nutrisi dan harga dapat dilihat pada Lampiran A).

Hasil komposisi untuk tiap-tiap bahan pakan yang dihasilkan oleh MSPSO dapat dilihat pada Tabel 5.15. Warna merah pada komposisi menunjukkan nilai komposisi yang melewati batasan maksimum bahan pakan tersebut. Sedangkan warna biru menunjukkan komposisi yang tidak digunakan sama sekali dalam formulasi atau sama dengan nol.

**Tabel 5.13** Kompisisi tiap-tiap bahan dan harga per kilogram

Bahan Pakan	Komposisi	Batasan
DEDAK HALUS	4,311%	40%
JAGUNG KUNING	42,666%	60%
BUNGKIL KEDELE	15,547%	40%
BUNGKIL KELAPA	25,096%	25%
T.IKAN (Menhaden)	0,000%	10%
TEPUNG DAGING	6,557%	7%
TEPUNG TULANG	5,823%	5%
MINYAK KELAPA	0,000%	5%
<b>Biaya / Kg.</b>	<b>Rp. 4.263,0759</b>	

Hasil komposisi tersebut menunjukkan terapat dua bahan pakan yang melewati batas maksimum dan tidak digunakan sama sekali pada formula. Bahan pakan yang melanggar batas maksimum adalah bungkil kelapa dan tepung tulang yang masing-masing melewati batas maksimum sebesar 0,96% dan 0,823%. Menurut pakar pakan ternak, komposisi yang terlanggar tersebut masih dapat ditoleransi. Sedangkan tepung ikan (menhaden) dan minyak kelapa tidak digunakan sama sekali yang menunjukkan bahwa bahan pakan tersebut dapat diganti dengan bahan pakan yang lain agar formula yang dihasilkan memiliki penalti yang lebih kecil dengan harga yang lebih murah.

Total penalti nutrisi dan penalti batasan komposisi dapat dilihat pada Tabel 5.16. Penalti nutrisi non asam amino adalah sebesar 4,884 dan penalti nutrisi asam amino adalah sebesar 0,323 dengan total keseluruhan penalti nutrisi sebesar 5,207. Kandungan nutrisi yang dilanggar untuk non asam amino dan asam amino dijabarkan lebih rinci pada Tabel 5.25 dan 5.26. Sedangkan penalti batasan komposisi dapat ditoleransi sebesar 0,687.

**Tabel 5.14** Penalti yang diperoleh untuk fase *layer*

Penalti	Besaran
Total Penalti Nutrisi	5,207
Nutrisi Non Asam Amino	4,884
Keseimbangan Asam Amino	0,323
Batasan Komposisi	0,687

Rincian penalti nutrisi non asam amino dapat dilihat pada Tabel 5.25. Keterangan dengan *font* warna hijau menunjukkan terpenuhinya nutrisi sesuai kebutuhan pada fase *layer*, warna biru menunjukkan adanya kandungan nutrisi yang melebihi batasan maksimum, dan warna merah menunjukkan adanya kandungan nutrisi yang kurang dari batasan minimum. Seperti terlihat pada tabel tersebut, hanya terdapat tiga nutrisi yang memenuhi batasan minimum dan maksimum kebutuhan nutrisi yang dianjurkan pada fase *layer* yaitu metabolisme energi (EM), mangan (Mn), dan zinc (Zn). Kandungan nutrisi yang melanggar batasan dibahas sebagai berikut :

**Tabel 5.15 Status terpenuhi dan penalti untuk kebutuhan nutrisi non asam amino pada fase *layer***

Nutrisi	Nilai	Kebutuhan	Keterangan
EM	2,650	2450 - 2650	Terpenuhi
PK	19,597	17 - 19	+0,597
LK	3,187	4 - 6	-0,813
SK	7,379	4 - 6	+1,379
Ca	2,014	3,3 - 3,8	-1,286
P	1,082	0,7 - 0,9	+0,182
Na	0,026	0,09 - 0,15	-0,064
K	0,655	0,09 - 0,15	+0,505
Cl	0,032	0,09 - 0,15	-0,058
Mn	30,000	25 - 30	Terpenuhi
Zn	34,853	30 - 35	Terpenuhi

Kandungan protein kasar (PK) pada formula tersebut melebihi batas maksimum dengan kelebihan sebesar 0,597 atau 3,14% dari batas maksimum sebesar 19. Namun, kandungan protein kasar tersebut dapat ditoleransi menurut pakar pakan ternak dan keseimbangan asam amino yang menyusun protein tersebut perlu diperhatikan.

Pada lemak kasar (LK), kandungan nutrisinya kurang dari kebutuhan minimum dengan kekurangan sebesar 0,813 atau 20,33% dari batas minimum sebesar 4. Lemak ditambahkan pada pakan dengan tujuan untuk meningkatkan konsentrasi keseluruhan energi yang berasosiasi dengan produktifitas (National Research Council, 1994) dan menurut pakar, kekurangan tersebut dapat ditoleransi.

Kandungan serat kasar (SK) didapati melewati batas maksimum dengan kelebihan sebesar 1,379 atau 22,98%. Kandungan serat kasar tersebut masih dapat ditoleransi menurut pakar pakan ternak.

Kandungan kalsium (Ca) yang diperoleh adalah sebesar 2,014 dengan kekurangan nutrisi sebesar 1,286 atau 38,97% dari batas minimum sebesar 3,3. Kekuatan cangkang telur akan meningkat seiring dengan peningkatan kandungan kalsium (National Research Council, 1994) dan kandungan tersebut dapat ditoleransi menurut pakar pakan ternak.

Kandungan fosfor yang terdapat pada formula tersebut adalah sebesar 1,082 dengan kelebihan nutrisi sebesar 0,182 atau sebesar 20,2. Asupan fosfor yang berlebih dapat mengurangi kekuatan cangkang telur (National Research Council, 1994). Namun, menurut pakar pakan ternak kandungan nutrisi tersebut dapat ditoleransi.

Kandungan natrium (Na) pada formula tersebut adalah sebesar 0,026 dengan kekurangan kandungan nutrisi sebesar 0,064 atau 71% dari batas minimum sebesar 0,09. Produksi telur cenderung menurun dengan berkurangnya kandungan nutrisi natrium. Hasil formulasi tersebut menurut pakar dapat ditangani dengan menambah garam untuk dapat meningkatkan kandungan natrium (National Research Council, 1994)

Selanjutnya, kandungan kalium (K) melebihi batas maksimum kebutuhan nutrisi yaitu sebesar 0,505 atau 336,67% dari batas maksimum sebesar 0,15. Penalti tersebut menunjukkan formula dengan diet kalium yang berlebih. Menurut pakar kandungan kalium tersebut masih dapat ditoleransi.

Nutrisi terakhir yang tidak memenuhi kebutuhan fase *layer* adalah *chlorine* (Cl). Kandungan nutrisi tersebut kurang dari batas minimum yang dianjurkan yaitu sebesar 0,058 atau 64,44% dari batas minimum pada fase *layer* yaitu sebesar 0,09. Namun, menurut pakar kurangnya kalium tersebut dapat ditoleransi.

Sedangkan kandungan nutrisi asam amino secara rinci dijabarkan pada Tabel 5.18. Hampir semua kandungan nutrisi asam amino melebihi batas minimum dari ayam petelur pada fase *layer* kecuali *leucine* (Leu). Kandungan asam amino *leucine* adalah sebesar 0,997 yang kurang dari kebutuhan minimum sebesar 1,32 sehingga daya cerna hanya sebesar 0,997. Diperlukan *leucine* sebesar 0,323 untuk bisa mencapai batas minimum kebutuhan asam amino tersebut yang menurut pakar dapat ditoleransi.

**Tabel 5.16 Status terpenuhi dan penalti untuk keseimbangan asam amino pada fase *layer***

Nutrisi	Nilai	Kebutuhan	Daya Cerna	Keterangan
Arg	1,449	0,85	0,850	Terpenuhi (+0,599)
Cys	0,273	0,27	0,270	Terpenuhi (+0,003)
Gly	0,783	0,5	0,500	Terpenuhi (+0,283)
His	0,421	0,34	0,340	Terpenuhi (+0,081)
Isol	0,751	0,68	0,680	Terpenuhi (+0,071)
Leu	0,997	1,32	0,997	-0,323
Lis	0,730	0,73	0,730	Terpenuhi
Met	0,554	0,34	0,340	Terpenuhi (+0,214)
Fenil	0,897	0,78	0,780	Terpenuhi (+0,117)
Thre	0,640	0,45	0,450	Terpenuhi (+0,190)
Trip	0,190	0,15	0,150	Terpenuhi (+0,040)
Tir	0,454	0,4	0,400	Terpenuhi (+0,054)
Val	0,813	0,68	0,680	Terpenuhi (+0,133)

Saran yang dapat diberikan dari hasil simulasi tersebut adalah bahan pakan tepung ikan (Menhaden) dan minyak kelapa dapat diganti dengan bahan pakan yang lain untuk bisa memenuhi kebutuhan ayam petelur pada fase *layer*. Sebagai contoh bahan pakan yang kaya akan natrium atau *leucine* dapat diberikan untuk menggantikan bahan pakan tersebut.

## 5.9 Pembahasan Hipotesis

H1 : Kualitas solusi dapat ditingkatkan seiring dengan peningkatan jumlah *sub-swarm*.

Pembahasan :

Berdasarkan hasil uji jumlah *sub-swarm* yang dibahas pada sub-bab 5.4, pada formula tertentu kenaikan jumlah *sub-swarm* dapat meningkatkan rata-rata *fitness* yang menunjukkan peningkatan kualitas solusi yang dihasilkan oleh hibridisasi PSO. Pada kasus formula yang lain, terjadi penurunan rata-rata *fitness*. Setelah jumlah *sub-swarm* dinaikkan pada kasus formula tersebut, rata-rata *fitness* menunjukkan adanya kenaikan. Hasil pengujian tersebut menunjukkan bahwa kualitas solusi dapat ditingkatkan seiring dengan peningkatan jumlah *sub-swarm* jika *swarm size* untuk tiap-tiap *sub-swarm* tidak kecil.

H2 : Jumlah *swarm size* dan iterasi yang optimum sangat dipengaruhi oleh kombinasi bahan pakan.

Pembahasan :

Berdasarkan hasil uji jumlah *swarm size* dan iterasi yang dibahas pada sub-bab 5.5 dan 5.6, jumlah *swarm size* dan iterasi yang optimum menunjukkan hasil yang berbeda-beda pada tiap-tiap formula. Hal tersebut menunjukkan bahwa *swarm size* dan iterasi yang optimum dependen terhadap kombinasi bahan pakan. Sehingga jumlah *swarm size* dan iterasi yang optimum sangat dipengaruhi oleh kombinasi bahan pakan.

H3 : Dengan meningkatkan keberagaman partikel pada hibridisasi PSO melalui *multi-swarm* dan komunikasi antar partikel melalui biseksi, hibridisasi PSO mampu menghasilkan solusi yang lebih baik dibandingkan dengan algoritme genetika, *particle swarm optimization*, dan hibridisasi algoritme genetika adaptif dengan *simulated annealing*.

Pembahasan :

Berdasarkan hasil uji komparasi yang dibahas pada sub-bab 5.8. Rata-rata *fitness* yang dihasilkan oleh hibridisasi PSO dapat memberikan nilai yang lebih tinggi dibandingkan dengan algoritme komparasi lainnya. Terlebih saat model tersebut diuji pada kebutuhan nutrisi sesuai Standar Nasional Indonesia (SNI), model hibridisasi PSO dapat memberikan solusi dengan penalti mendekati nol namun dengan biaya yang rendah dibandingkan dengan algoritme komparasi



lainnya. Hal tersebut menunjukkan bahwa kualitas solusi yang dihasilkan oleh hibridisasi PSO lebih baik dibandingkan dengan algoritme genetika, *particle swarm optimization*, dan hibridisasi algoritme genetika adaptif dengan *simulated annealing*.