

BAB 4 PERANCANGAN

Bab ini membahas secara rinci model matematis dalam menentukan proporsi bahan pakan ayam petelur. Selanjutnya model hibridisasi PSO dibahas lebih rinci yang meliputi representasi partikel, inialisasi partikel, fungsi *fitness*, dan model hibridisasi PSO. Untuk mengetahui efektivitas model hibridisasi PSO yang memanfaatkan *multi-swarm*, skenario pengujian dirancang lebih rinci. Berikut sub-bab yang membahas model matematis, model hibridisasi PSO, dan skenario pengujian.

4.1 Model Matematis

Dalam dunia nyata untuk menyiapkan bahan pakan yang akan dicampurkan perlu diketahui berat pakan yang diperlukan untuk diberikan pada hewan ternak. Berat pakan tersebut bervariasi tergantung dari jumlah hewan ternak yang dimiliki. Jika kombinasi bahan pakan yang dicampurkan sama namun memerlukan berat yang berbeda, maka hasil proporsi berat masing-masing bahan pakan akan berbeda. Sehingga untuk menghasilkan formula yang sama dengan kombinasi bahan pakan yang sama, persentase tiap-tiap bahan pakan dapat digunakan. Dengan menggunakan persentase, proporsi tiap-tiap bahan pakan tetap dan dapat diproyeksikan dengan berat pakan yang berbeda-beda sesuai kebutuhan peternak. Oleh sebab itu, penelitian ini menggunakan persentase sebagai komposisi tiap-tiap bahan pakan.

4.1.1 Kendala Jumlah Komposisi Pakan

Jika himpunan G menyatakan daftar bahan pakan yang dinyatakan pada Persamaan 4.1 dan barisan H merupakan bahan pakan yang dipilih yang memiliki indeks i sesuai urutan bahan pakan yang dinyatakan pada Persamaan 4.2, maka komposisi tiap-tiap anggota pada barisan H dapat dinyatakan sebagai vektor \bar{X} seperti terlihat pada Persamaan 4.3.

$$G = \{g \mid g \text{ adalah bahan pakan}\} \quad (4.1)$$

$$H = \langle h_i \mid h \text{ adalah bahan pakan yang dipilih, } i \in I, h_i \in G \rangle \quad (4.2)$$

$$\bar{X} = (x_1, x_2, \dots, x_i, \dots, x_{|H|}), \forall x \in \mathbb{R}_{>0} \quad (4.3)$$

Komposisi bahan pakan yang berupa persentase mengharuskan jumlah semua bahan pakan sama dengan 100%. Sebagai contoh bahan pakan $X = 20\%$, $Y = 50\%$, dan $Z = 30\%$ memiliki jumlah keseluruhan bahan pakan adalah 100%. Sehingga, persentase masing-masing bahan pakan merupakan *hard constraint* yang harus dipenuhi. Jika \bar{X} merupakan vektor yang terdiri dari komposisi untuk semua anggota himpunan H atau pilihan bahan pakan yang dapat dinyatakan pada Persamaan 4.3, maka *hard constraint* dapat dinyatakan pada Persamaan 4.4. Panjang vektor \bar{X} sama dengan kardinalitas pada himpunan H dan x_i merupakan persentase bahan pakan ke- i yang merujuk pada urutan di anggota himpunan H

atau bahan pakan h . Sedangkan $|H|$ merupakan kardinalitas atau jumlah keseluruhan anggota himpunan H atau pilihan bahan pakan.

$$\sum_{i=1}^{|H|} x_i = 100\% \quad (4.4)$$

Persentase tiap-tiap bahan pakan harus menghasilkan jumlah zat nutrisi yang memenuhi kebutuhan ayam petelur dan seimbangnya asam amino. Jika jumlah zat nutrisi tidak memenuhi kebutuhan ayam petelur maka akan menurunkan produksi telur dan jika asam amino tidak seimbang maka asam amino yang ada dalam pakan tidak dapat dicerna secara sempurna sehingga akan mempengaruhi produksi telur ayam. Selain itu tiap-tiap bahan pakan memiliki batasan maksimal persentase yang sebaiknya dipenuhi. Oleh sebab itu, penalti akan diberikan jika zat nutrisi pakan melanggar batasan-batasan nutrisi yang harus dipenuhi dan melanggar persentase batas maksimal bahan pakan yang dianjurkan. Dengan diberikannya penalti, kebutuhan nutrisi, keseimbangan asam amino, dan maksimal persentase bahan pakan menjadi *soft constraint* yang sebaiknya dipenuhi. Sehingga tujuan yang ingin dicapai meliputi rendahnya penalti untuk ketiga kendala tersebut dan rendahnya harga pakan. Berikut dibahas secara rinci kendala kebutuhan nutrisi, kendala keseimbangan asam amino, kendala bahan pakan (secara rinci batasan bahan pakan dapat dilihat pada Lampiran C), dan fungsi tujuan.

4.1.2 Kendala Kebutuhan Nutrisi Non Asam Amino

Setiap periode umur tertentu, ayam petelur memerlukan batasan nutrisi yang berbeda-beda yang memiliki batasan minimum dan maksimum. Sehingga kebutuhan nutrisi terpenuhi jika semua zat nutrisi pakan berada pada rentang minimum dan maksimum dari kebutuhan nutrisi tersebut. Kumpulan nutrisi dipertimbangkan dapat dinyatakan dalam himpunan A yang dapat dilihat pada Persamaan 4.5. Nutrisi yang dipertimbangkan dalam penelitian ini antara lain nutrisi non asam amino yang meliputi energi metabolisme (me), protein kasar (cp), lemak kasar (f), serat kasar (cf), kalsium (ca), *phosphorus* (p), natrium (na), *potassium* (k), *chlorine* (cl), *manganese* (mn), *zinc* (zn) dan asam amino yang terdiri dari *arginine* (arg), *cystine* (cys), *glycine* (gly), *histidine* (his), *isoleucine* ($isol$), *leucine* (leu), *lysine* (lis), *methionine* (met), *phenylalanine* (*fenil*), *threonine* ($thre$), *tryptophan* ($trip$), *tyrosine* (tir), dan *valine* (val).

$$A = \{me, cp, \dots, lis, \dots, val\} \quad (4.5)$$

Kebutuhan nutrisi ayam petelur berbeda menurut fase umur yang dapat dibagi menjadi fase *starter*, *grower*, dan *layer* yang dapat dinyatakan dalam himpunan F seperti terlihat pada Persamaan 4.6. Setiap anggota himpunan F dan semua anggota himpunan A dapat dipetakan kedalam bilangan riil positif yang dapat dinotasikan pada Persamaan 4.7. Pemetaan tersebut menunjukkan adanya relasi antara himpunan A dan F terhadap bilangan riil positive yang dalam kasus pakan ternak bilangan tersebut merujuk pada nilai minimum dan maksimum kebutuhan nutrisi ayam petelur. Fungsi minimum dan maksimum kebutuhan nutrisi dapat dinyatakan pada Persamaan 4.8 dan 4.9.

$$F = \{starter, grower, layer\} \quad (4.6)$$

$$f: A, F \rightarrow \mathbb{R}_{>0} \quad (4.7)$$

$$\min(a, f), a \in A, f \in F \quad (4.8)$$

$$\max(a, f), a \in A, f \in F \quad (4.9)$$

Nutrisi yang dipertimbangkan dibagi menjadi dua yaitu, nutrisi non asam amino dan nutrisi asam amino. Non asam amino, seperti metabolisme energi, protein kasar, serat kasar, lemak kasar, dan lain-lain yang telah dibahas pada uraian sebelumnya dapat dibangun pada himpunan B seperti terlihat pada Persamaan 4.10. Himpunan tersebut sekaligus menunjukkan bahwa B merupakan himpunan bagian dari himpunan A yang dinyatakan pada Persamaan 4.11.

$$B = \{b \mid b \text{ adalah nutrisi non asam amino}\} \quad (4.10)$$

$$B \subset A \quad (4.11)$$

Setiap himpunan anggota A atau nutrisi dengan bahan pakan dipetakan ke bilangan riil positif yang menunjukkan fungsi berupa kandungan nutrisi anggota himpunan A dalam 1 kg bahan pakan seperti terlihat pada Persamaan 4.12. Sehingga baik nutrisi non asam amino ataupun nutrisi asam amino memiliki kandungan nutrisi yang dapat diperoleh melalui fungsi pada Persamaan 4.13.

$$f: A, G \rightarrow \mathbb{R}_{>0} \quad (4.12)$$

$$\text{nut}(a, g), a \in A, g \in G \quad (4.13)$$

Jika $tN_b(\bar{X})$ menunjukkan total kandungan nutrisi b yang terkandung dalam \bar{X} yang dapat dinyatakan pada Persamaan 4.14, maka kendala kebutuhan nutrisi dapat dinyatakan pada Persamaan 4.15. Kendala tersebut menunjukkan bahwa total nutrisi b dari pakan harus berada pada rentang kebutuhan nutrisi minimum b dan maksimum b .

$$tN_b(\bar{X}) = \sum_{i=1}^{|H|} \frac{x_i}{100} \times \text{nut}(b, h_i) \quad (4.14)$$

$$\min(b, f) \leq tN_b(\bar{X}) \leq \max(b, f) \quad (4.15)$$

4.1.3 Kendala Keseimbangan Asam Amino

Asam amino merupakan salah satu bagian dari kebutuhan nutrisi yang dipertimbangkan yang dapat dinyatakan sebagai himpunan C seperti terlihat pada Persamaan 4.16. Sehingga C merupakan himpunan bagian dari himpunan A seperti terlihat pada Persamaan 4.17.

$$C = \{c \mid c \text{ adalah nutrisi asam amino}\} \quad (4.16)$$

$$C \subset A \quad (4.17)$$

Keseimbangan asam amino diperlukan jika total nutrisi *lis* pada pakan kurang dari kebutuhan yang diperlukan sehingga mempengaruhi nutrisi yang dapat dicerna oleh ayam petelur dan menjadi kurang optimal. Zat nutrisi asam amino yang dapat dicerna dapat dihitung dari kebutuhan asam amino yang relatif terhadap nilai *lis* dari total keseluruhan kandungan asam amino tersebut jika total nutrisi *lis* kurang dari kebutuhan. Sebaliknya jika total nutrisi *lysine* sudah terpenuhi maka semua nutrisi asam amino dapat dicerna sesuai dengan daya cernanya.

Untuk semua kemungkinan dari total nutrisi asam amino *c* yang seimbang (dN_c) dapat dinyatakan pada Persamaan 4.18. Persamaan tersebut menunjukkan bahwa jika total nutrisi *lis* kurang dari kebutuhan dan total nutrisi *c* pada pakan melebihi daya cerna nutrisi *c* (D_c), maka nutrisi yang dapat dicerna adalah $D_c(\bar{X})$ yang dinyatakan pada Persamaan 4.19. Jika total nutrisi *lis* melebihi batas minimum kebutuhan, maka nutrisi asam amino yang terkandung dalam pakan dapat dicerna walaupun tidak sepenuhnya (misal jika nutrisi *c* kurang dari kebutuhan maka nutrisi *c* dapat dicerna seutuhnya tetapi jika melebihi kebutuhan maka hanya batas minimal kebutuhan yang dapat dicerna). Begitu juga jika total nutrisi *lis* kurang dari kebutuhan minimum *lis* dan total nutrisi *c* kurang dari daya cerna nutrisi *c*. Sehingga hanya total nutrisi *c* yang dapat dicerna.

$$dN_c(\bar{X}) = \begin{cases} D_c(\bar{X}), & tN_{lis}(\bar{X}) < \min(lis, f) \wedge tN_c(\bar{X}) \geq D_c(\bar{X}) \\ tN_c(\bar{X}), & tN_{lis}(\bar{X}) \geq \min(lis, f) \vee () \\ tN_c(\bar{X}), & tN_{lis}(\bar{X}) < \min(lis, f) \wedge tN_c(\bar{X}) < D_c(\bar{X}) \end{cases} \quad (4.18)$$

$$D_c(\bar{X}) = \frac{\min(c, f)}{\min(lis, f)} \times tN_{Lys}(\bar{X}) \quad (4.19)$$

Hasil dari keseimbangan asam amino *c* pada persamaan tersebut harus memenuhi kendala kebutuhan nutrisi *c* ayam petelur. Pada kenyataannya, tidak ada batas maksimum yang dianjurkan untuk nutrisi asam amino. Karena nutrisi yang paling baik sebaiknya mendekati nilai minimum atau dapat dikatakan sama dengan kebutuhan minimum. Nilai nutrisi asam amino yang berlebihan akan berdampak pada tingkat ketajaman aroma fekes ayam petelur. Sehingga kendala nutrisi asam amino dapat dinyatakan pada Persamaan 4.20.

$$\min(c, f) = dN_c(\bar{X}) \quad (4.20)$$

4.1.4 Kendala Bahan Pakan

Tiap-tiap bahan pakan memiliki batasan maksimum komposisi yang dianjurkan oleh pakar peternakan. Bahan pakan yang dipilih dan fase umur ayam petelur dapat dipetakan kedalam bilangan riil positif seperti terlihat pada Persamaan 4.21 yang sekaligus berfungsi dalam pemetaan nilai batasan maksimum komposisi dari bahan pakan *h* pada fase *f* seperti terlihat pada Persamaan 4.22. Jika $\max(h, f)$ menyatakan batasan maksimum pada bahan pakan *h*, maka kendala bahan pakan yang harus dipenuhi dapat dinyatakan pada Persamaan 4.23.

$$f: H, F \rightarrow \mathbb{R}_{>0} \quad (4.21)$$

$$\max(h, f), h \in H, f \in F \quad (4.22)$$

$$0 \leq x_h \leq \max(h, f) \quad (4.23)$$

4.1.5 Fungsi Tujuan

Tujuan dari optimasi formulasi pakan ayam petelur adalah untuk meminimalkan harga dan penalti dari *soft constraint* yang berupa kendala kebutuhan nutrisi, kendala keseimbangan asam amino, dan kendala bahan pakan. Sehingga fungsi tujuan adalah minimnya harga dan penalti dari formula pakan.

Fungsi tujuan dapat dilihat pada Persamaan 4.24. Pada persamaan tersebut c_i merupakan harga bahan pakan ke- i . $pN(\bar{X})$ merupakan nilai yang diperoleh saat kandungan nutrisi pakan diluar batasan kebutuhan nutrisi non asam amino (penalti kebutuhan nutrisi non asam amino) yang dinyatakan pada Persamaan 4.25, $pAA(\bar{X})$ merupakan nilai yang diperoleh saat nilai keseimbangan asam amino diluar dari batasan kebutuhan nutrisi (penalti keseimbangan asam amino) yang dinyatakan pada Persamaan 4.26, dan $pMI(\bar{X})$ merupakan jumlah nilai dari tiap-tiap bahan pakan yang melewati batas maksimum tiap-tiap bahan pakan (penalti bahan pakan) yang dinyatakan pada Persamaan 4.27.

$$\min \sum_{i=1}^{|H|} (c_i \times x_i) + pN(\bar{X}) + pAA(\bar{X}) + pMI(\bar{X}) \quad (4.24)$$

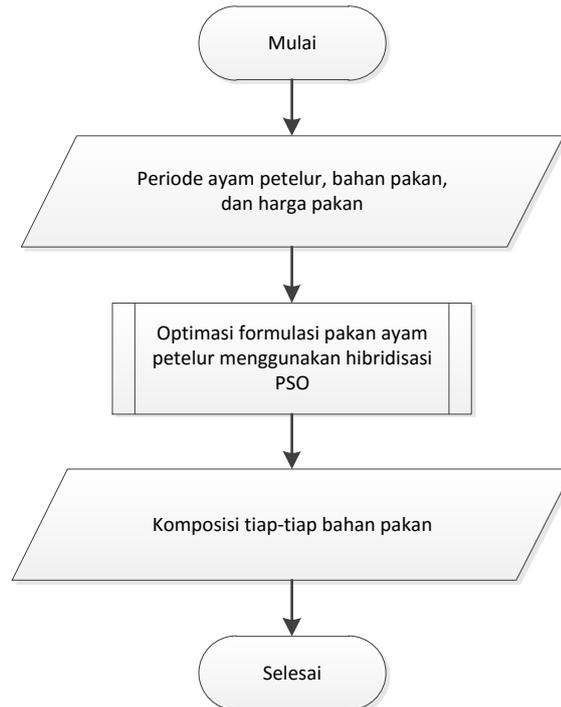
$$pN(\bar{X}) = \sum_{b \in B} \begin{cases} tN_b(\bar{X}) - \max(b, f), & \max(b, f) < tN_b(\bar{X}) \\ \min(b, f) - tN_b(\bar{X}), & \min(b, f) > tN_b(\bar{X}) \\ 0, & \min(b, f) \leq tN_b(\bar{X}) \leq \max(b, f) \end{cases} \quad (4.25)$$

$$pAA(\bar{X}) = \sum_{c \in C} \begin{cases} \min(c, f) - dN_c(\bar{X}), & dN_c(\bar{X}) < \min(c, f) \\ 0, & dN_c(\bar{X}) \geq \min(c, f) \end{cases} \quad (4.26)$$

$$pMI(\bar{X}) = \sum_{i=1}^{|H|} \begin{cases} x_i - \max(h_i, f), & \max(h_i, f) < x_i \\ 0, & \max(h_i, f) \geq x_i \end{cases} \quad (4.27)$$

4.2 Optimasi Formulasi Pakan Ayam Petelur

Rancangan optimasi formulasi pakan ayam petelur secara umum terdiri dari masukan dan keluaran proses optimasi yang dapat dilihat pada Gambar 4.1. Masukan dari optimasi tersebut adalah berupa periode ayam petelur, bahan-bahan pakan yang akan dipertimbangkan dalam formulasi, dan harga bahan pakan itu sendiri yang berbeda-beda di tiap daerah dengan harga yang fluktuatif. Masukan tersebut akan diproses dalam model optimasi menggunakan hibridisasi PSO lalu menghasilkan keluaran berupa komposisi tiap-tiap bahan pakan yang sudah mendekati atau terpenuhinya kendala kebutuhan nutrisi, kendala keseimbangan asam amino, dan kendala bahan pakan dengan harga yang minimal.



Gambar 4.1 Diagram alir optimasi formulasi pakan ayam petelur

Suatu permasalahan dalam dunia nyata yang akan dioptimasi harus dapat direpresentasikan kedalam model optimasi tersebut. Dalam PSO, suatu kandidat atau contoh solusi suatu formula direpresentasikan kedalam suatu partikel yang akan dibangkitkan secara random. Kandidat solusi tersebut mula-mula merupakan solusi yang tidak optimal yang diukur melalui fungsi *fitness*. Pada tiap iterasinya, partikel akan dipindah posisikan berdasarkan komponen akselerasinya menuju posisi yang lebih baik. Oleh sebab itu, sebelum perancangan model hibridisasi PSO dirancang, komponen-komponen yang ada pada PSO harus dapat merepresentasikan masalah dunia nyata seperti representasi partikel, proses inialisasi partikel awal, dan fungsi *fitness* akan dirancang dan dibahas lebih rinci pada sub-bab berikut.

4.3 Representasi Partikel

Partikel pada PSO sejatinya merupakan vektor dengan bilangan riil. Representasi tersebut akan membuat komposisi bahan pakan menjadi lebih presisi. Sehingga representasi PSO yang digunakan untuk merepresentasikan komposisi tiap-tiap bahan pakan adalah representasi riil. Panjang dimensi suatu partikel menunjukkan banyaknya bahan pakan yang akan dioptimasi dimana tiap-tiap dimensi merupakan bilangan riil.

Suatu partikel memiliki posisi masing-masing yang menunjukkan komposisi bahan pakan yang merupakan bilangan riil seperti terlihat pada Persamaan 4.28. Posisi tersebut akan bergerak berdasarkan kecepatan yang dimiliki suatu partikel yang dinyatakan pada Persamaan 4.29 yang akan diperbarui setiap iterasinya menggunakan Persamaan 2.9. Partikel ke-*i* pada iterasi ke-*t* dapat dinyatakan pada

Persamaan 4.30. Setiap partikel ke- i memiliki posisi yang dinyatakan pada \bar{X}_i yang sekaligus merepresentasikan komposisi bahan pakan dan memiliki kecepatan yang dinyatakan pada \bar{V}_i . $x_{i,1}$ menunjukkan posisi partikel ke- i pada dimensi pertama dan $v_{i,1}$ menunjukkan kecepatan partikel ke- i pada dimensi pertama. Sedangkan D menunjukkan total dimensi yang digunakan yang sekaligus merepresentasikan jumlah bahan pakan yang dioptimasi.

$$\bar{X}_i = (x_{i,1}, \dots, x_{i,j}, \dots, x_{i,D}), D = |H|, \forall x_i \in \mathbb{R} \quad (4.28)$$

$$\bar{V}_i = (v_{i,1}, \dots, v_{i,j}, \dots, v_{i,D}), D = |H|, \forall v_i \in \mathbb{R} \quad (4.29)$$

$$P_i(t) = \{\bar{X}_i, \bar{V}_i\} \quad (4.30)$$

Partikel tersebut harus dapat memenuhi *hard constraint* yang telah diuraikan pada sub-bab 4.1 dimana jumlah seluruh posisi pada partikel harus sama dengan 100%. Selama proses pergerakan partikel tiap iterasinya, posisi tiap-tiap dimensi dari suatu partikel memungkinkan jumlah semua posisi tidak sama dengan 100%. Sehingga partikel tersebut memerlukan perbaikan atau penyesuaian agar sama dengan 100%. Proses perbaikan tersebut akan diperbaiki sehingga jumlah semua posisi menjadi 100% dengan menggunakan Persamaan 4.31. $x_{i,j}$ menunjukkan posisi pada partikel ke- i dan dimensi ke- j . Sedangkan D merupakan jumlah dimensi atau jumlah bahan pakan yang dioptimasi oleh PSO. Namun persamaan tersebut hanya dapat digunakan ketika semua nilai posisi berupa bilangan positif. Jika nilai negatif ditemukan, persamaan tersebut tidak akan menghasilkan solusi yang layak dan tidak memenuhi kendala bahan pakan yang telah didefinisikan pada Persamaan 4.4. Untuk menangani perbaikan partikel yang memiliki nilai negatif, nilai *fitness* negatif diberikan pada partikel tersebut yang akan dibahas secara rinci pada sub-bab 4.4.

$$x_{i,j} = \frac{x_{i,j}}{\sum_{j=1}^D x_{i,j}} \times 100\% \quad (4.31)$$

4.4 Fungsi *Fitness*

Fungsi *fitness* merupakan suatu fungsi untuk mengevaluasi suatu partikel dengan penilaian tertentu. Semakin tinggi nilai *fitness* suatu partikel menunjukkan semakin baik partikel tersebut. Fungsi *fitness* didefinisikan pada Persamaan 4.32. Jika semua komposisi bahan pakan positif, maka nilai *fitness* menjadi 1 dibagikan dengan penjumlahan antara harga bahan pakan yang telah dinormalisasi $nCost(\bar{X}_i)$, penalti kebutuhan nutrisi $pN(\bar{X}_i)$, penalti keseimbangan asam amino $pAA(\bar{X}_i)$, dan penalti komposisi bahan pakan $pBP(\bar{X}_i)$. Jika selama pergerakan partikel terdapat posisi yang negatif maka partikel tersebut tidak dapat merepresentasikan solusi yang layak. Untuk mengatasi hal tersebut nilai *fitness* negatif dari jumlah semua posisi yang negatif diberikan pada partikel tersebut. Dengan penanganan posisi negatif dari suatu partikel melalui fungsi *fitness*, hal ini membuat partikel tersebut dipertimbangkan dalam proses pergerakan partikel.

$$F(\bar{X}_i) = \begin{cases} \frac{1}{nCost(\bar{X}_i) + pN(\bar{X}_i) + pAA(\bar{X}_i) + pMI(\bar{X}_i)}, & \forall x \in \bar{X}_i: x \geq 0 \\ \sum_{j=1}^D \begin{cases} x_{i,j}, & x_{i,j} < 0 \\ 0, & x_{i,j} \geq 0 \end{cases}, & \exists x \in \bar{X}_i: x < 0 \end{cases} \quad (4.32)$$

Fungsi $nCost(\bar{X}_i)$ menunjukkan total harga pakan yang telah dinormalisasi. Harga diperlukan normalisasi agar nilai harga tersebut tidak terlalu jauh dengan nilai nutrisi sehingga nilai harga terhadap suatu partikel diharapkan dapat seimbang dengan tujuan yang lain. Persamaan 4.33 digunakan untuk menormalisasi total harga suatu partikel. $minCost$ dan $maxCost$ masing-masing merupakan harga minimum dan maksimum dari bahan pakan per kg. Jika diasumsikan harga bahan pakan X, Y, dan Z dalam satu kilogram masing-masing sebesar X = Rp. 5.000,00, Y = Rp. 3.000,00, dan Z = Rp. 1.000,00 maka $minCost$ adalah Rp. 1.000,00 dan $maxCost$ adalah Rp. 5.000,00. $tCost(P_i(t))$ menunjukkan total seluruh biaya pakan dalam 100 kilogram pada partikel ke- i dengan iterasi ke- t yang ditunjukkan pada Persamaan 4.34. Total biaya tersebut diperoleh dari penjumlahan perkalian antara komposisi pakan dengan biaya pakan per kilogram. c_j menunjukkan harga pada bahan pakan ke- j atau dimensi ke- j .

$$nCost(\bar{X}_i) = \frac{tCost(\bar{X}_i) - 100 \cdot minCost}{100 \cdot maxCost - 100 \cdot minCost} \quad (4.33)$$

$$tCost(\bar{X}_i) = \sum_{j=1}^D x_{i,j} \times c_j \quad (4.34)$$

Fungsi $pN(\bar{X}_i)$ menunjukkan nilai penalti dari kebutuhan nutrisi. Nilai penalti yang tinggi menunjukkan kerendahan kualitas calon solusi dimana banyak nilai-nilai nutrisi yang dilanggar dari kendala kebutuhan nutrisi. Fungsi $pN(\bar{X}_i)$ dapat dilihat pada Persamaan 4.35 yang merupakan penjumlahan dari total nutrisi suatu partikel yang melewati batas maksimum dan minimum kebutuhan nutrisi. $tN_b(\bar{X}_i)$ merupakan total nutrisi dari suatu partikel ke- i pada iterasi ke- t yang dapat dilihat pada Persamaan 4.14.

$$pN(\bar{X}_i) = \sum_{b \in B} \begin{cases} tN_b(\bar{X}_i) - \max(b, f), & \max(b, f) < tN_b(\bar{X}_i) \\ \min(b, f) - tN_b(\bar{X}_i), & \min(b, f) > tN_b(\bar{X}_i) \\ 0, & \min(b, f) \leq tN_b(\bar{X}_i) \leq \max(b, f) \end{cases} \quad (4.35)$$

Fungsi $pAA(\bar{X}_i)$ menunjukkan nilai penalti dari keseimbangan asam amino. Penalti akan diberikan jika suatu nutrisi asam amino yang seimbang melanggar batasan kebutuhan minimum dan maksimum nutrisi tersebut. Kebutuhan keseimbangan asam amino tidak memiliki rentang tertentu, melainkan diharapkan dapat mendekati nilai kebutuhan asam amino. Dalam hal ini, kebutuhan minimum sama dengan maksimum. Fungsi $pAA(\bar{X}_i)$ dapat dilihat pada Persamaan 4.36 yang merupakan penjumlahan dari keseimbangan asam amino yang melewati batasan minimum kebutuhan nutrisi asam amino tersebut. Penalti yang diberikan linear dengan rendahnya kualitas keseimbangan asam amino. Semakin tinggi penalti menunjukkan semakin sedikit nutrisi yang dapat dicerna. $dN_c(\bar{X}_i)$ menunjukkan nilai keseimbangan nutrisi asam amino c pada partikel ke- i yang dapat dicerna yang ditunjukkan pada Persamaan 4.18.

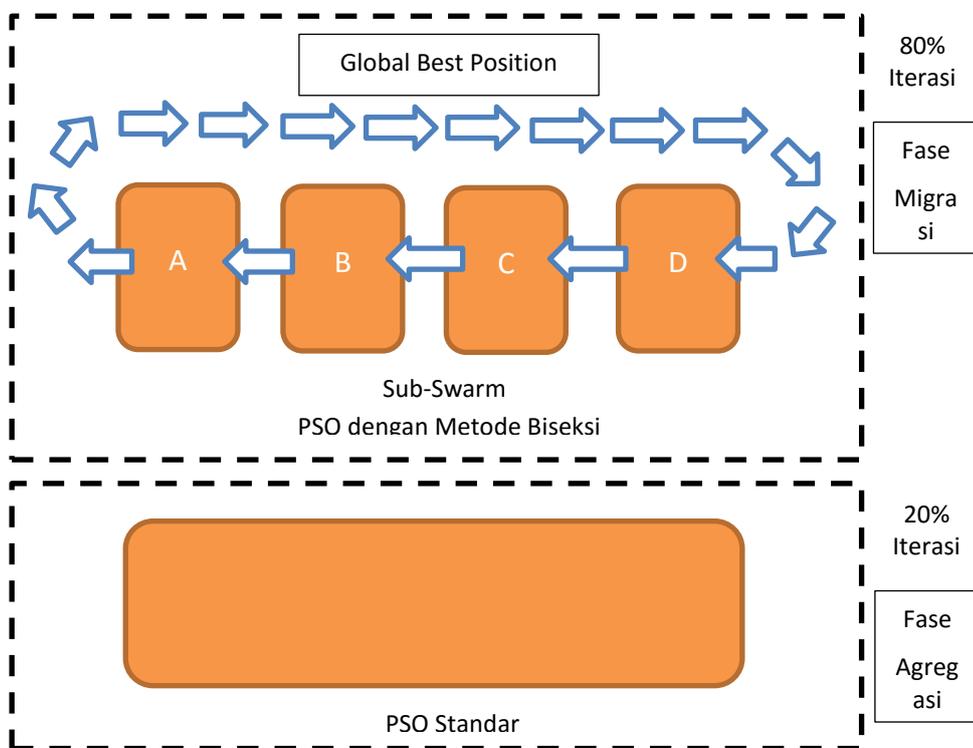
$$pAA(\bar{X}_i) = \sum_{c \in C} |\min(c, f) - dN_c(\bar{X}_i)| \quad (4.36)$$

Fungsi $pMI(\bar{X}_i)$ menunjukkan nilai penalti dari komposisi bahan pakan. Penalti tersebut diberikan jika komposisi bahan pakan melewati batas maksimal yang dianjurkan oleh pakar. Fungsi $pMI(\bar{X}_i)$ dapat dilihat pada Persamaan 4.37 yang diperoleh dari penjumlahan selisih antara batas maksimal dengan komposisi bahan pakan jika komposisi tersebut melewati batas maksimum.

$$pMI(\bar{X}_i) = \sum_{j=1}^D \begin{cases} x_{i,j} - \max(h_j, f), & \max(h_j, f) < x_{i,j} \\ 0, & \max(h_j, f) \geq x_{i,j} \end{cases} \quad (4.37)$$

4.5 Hibridisasi PSO

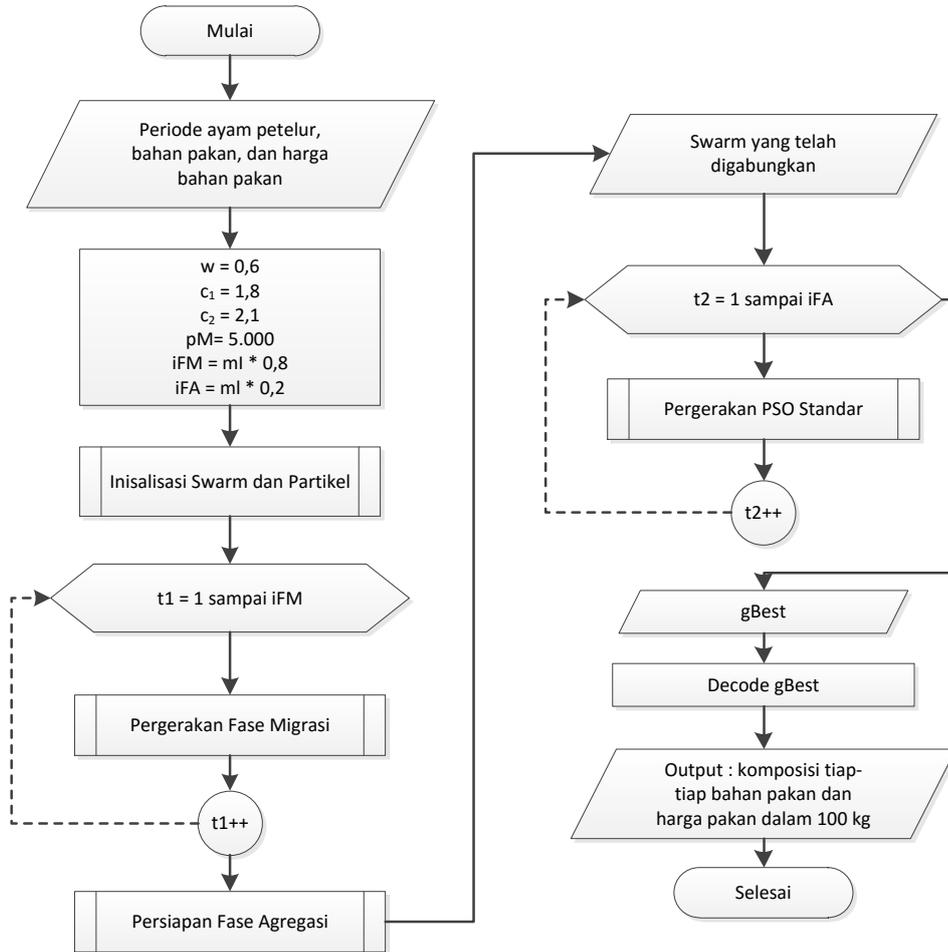
Representasi partikel, inisialisasi partikel dan fungsi *fitness* tersebut digunakan dalam rancangan hibridisasi PSO yang melibatkan *multi-swarm*. Model hibridisasi PSO dapat dilihat pada Gambar 4.2.



Gambar 4.2 Model hibridisasi PSO

Bentuk kotak pada gambar tersebut menunjukkan *sub-swarm*. *Swarm* pada PSO dibagi menjadi 4 *sub-swarm* yaitu A, B, C, dan D. Pergerakan partikel yang digunakan pada tiap-tiap *sub-swarm* telah ditingkatkan dengan probabilitas metode biseksi. Setiap periode tertentu posisi global terbaik dari masing-masing *sub-swarm* akan bermigrasi ke *sub-swarm* tetangganya seperti tergambar pada Gambar 4.2. Proses pergerakan tersebut bergerak dalam 80% iterasi yang telah ditentukan dan proses tersebut disebut sebagai fase migrasi. Selama 20% iterasi terakhir, semua *sub-swarm* bersatu menjadi satu *swarm* dan bergerak

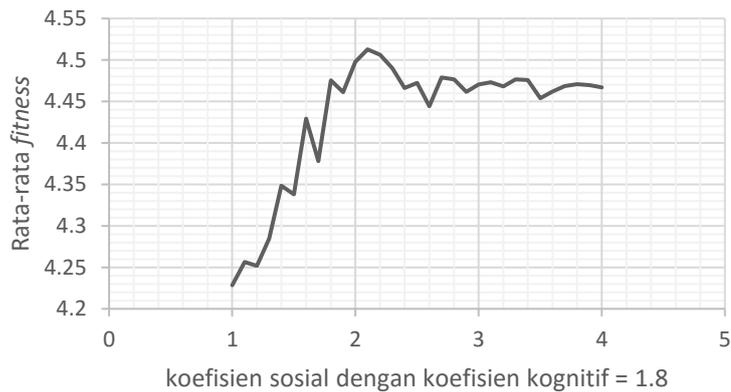
berdasarkan posisi global terbaik dengan menggunakan pergerakan partikel standar (PSO standar) sehingga pergerakan tersebut disebut sebagai fase agregasi.



Gambar 4.3 Diagram alir hibridisasi PSO

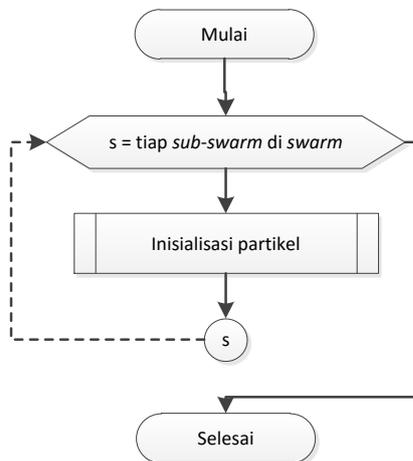
Secara rinci, hibridisasi PSO dirancang kedalam diagram alir yang tergambar pada Gambar 4.3. Masukan dari model hibridisasi PSO adalah berupa periode ayam petelur dimana periode tersebut berbeda-beda berdasarkan umur ayam petelur yang membutuhkan kebutuhan nutrisi yang berbeda-beda. Selain itu masukan adalah berupa bahan pakan yang akan dipilih oleh pengguna. Bahan pakan tersebut sudah memiliki nilai-nilai nutrisi yang tetap yang akan dipertimbangkan saat dipilih oleh pengguna. Harga pakan juga dapat berupa masukan dikarenakan harga di tiap daerah berbeda-beda dan fluktuatif. Ketiga masukan tersebut menjadi masukan yang akan diproses pada hibridisasi PSO.

Setelah masukan didapatkan, parameter PSO ditetapkan seperti pada Gambar 4.3. Parameter $w = 0,6$, $c_1 = 1,8$, dan $c_2 = 2,1$ diperoleh dari hasil percobaan yang menunjukkan bahwa parameter tersebut dapat menghasilkan solusi yang lebih baik dengan menggunakan beban inersia sebagai konstanta sebesar 0,6. Percobaan dilakukan secara enumerasi terhadap kedua koefisien akselerasi dari 1,0 sampai 4,0 dengan ketelitian satu dibelakang koma.



Gambar 4.4 Rata-rata *fitness* terhadap koefisien sosial dengan koefisien kognitif sebesar 1,8

Seperti terlihat pada Gambar 4.4, peningkatan koefisien sosial (c_2) dari 1 sampai 2,1 menunjukkan adanya peningkatan rata-rata *fitness*. Namun, peningkatan diatas 2,1 menurunkan performa PSO yang ditunjukkan dengan menurunnya rata-rata *fitness*. Sedangkan parameter pM merupakan periode migrasi yang ditetapkan sebagai konstanta sebesar 5,000. pM digunakan sebagai acuan periode dalam fase migrasi dimana posisi global terbaik akan bermigrasi ke *sub-swarm* terdekat. *iFM* dan *IFA* merupakan iterasi pada fase migrasi dan fase agregasi yang didapatkan dari 80% dan 20% dari maksimum iterasi (mI) pada hibridisasi PSO. Parameter iterasi dan *swarm size* pada PSO akan diuji untuk memperoleh jumlah iterasi dan *swarm size* yang baik. Parameter tersebut ditentukan setelah hasil pengujian dianalisis.



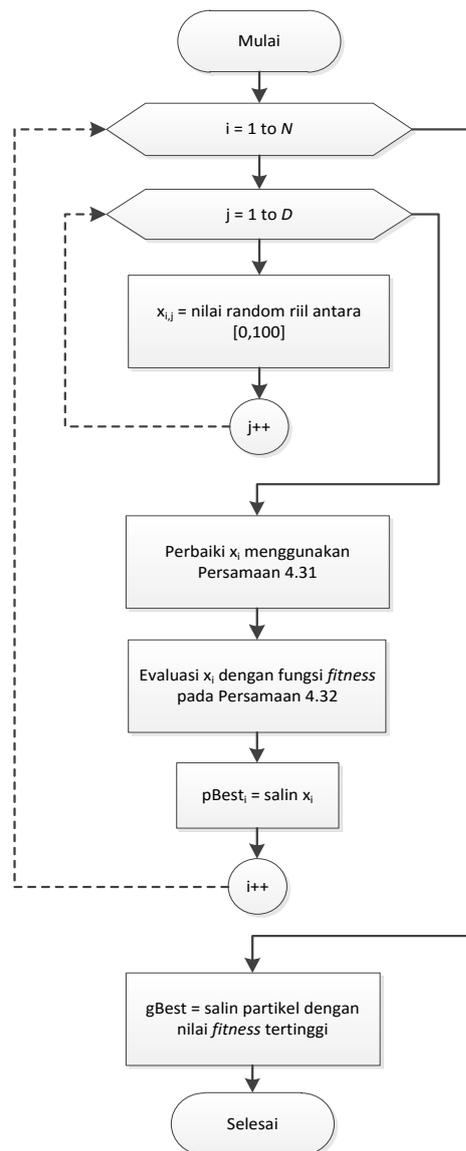
Gambar 4.5 Diagram alir inisialisasi *swarm* dan partikel

Setelah masukan didapatkan dan parameter di tentukan, masing-masing *sub-swarm* dan partikel di inisialisasi sebagai kandidat solusi awal. Secara rinci prosedur tersebut dibahas pada sub-bab 4.5.1. Setelah itu fase migrasi dijalankan selama *iFM* iterasi yang dibahas pada sub-bab 4.5.2. Selanjutnya semua *sub-swarm* digabungkan menjadi satu *swarm* yang menjadi persiapan fase agregasi yang dibahas pada sub-bab 4.5.3. Lalu *swarm* tersebut dijalankan dengan strategi

standar PSO selama *iFA* iterasi yang merupakan bagian dari fase agregasi yang dibahas pada sub-bab 4.5.5. Terakhir, *gBest* menjadi solusi terbaik yang ditemukan oleh hibridisasi PSO dan dilakukan *decode* untuk bisa diterjemahkan ke solusi dunia nyata yang berupa komposisi tiap-tiap bahan pakan.

4.5.1 Inisialisasi *Swarm* dan Partikel

Sub-swarm pada hibridisasi PSO ditetapkan sebanyak 4 swarm. Seperti terlihat pada Gambar 4.5. *Swarm* merupakan array yang terdiri dari *sub-swarm* sebanyak 4. Selanjutnya setiap partikel pada *sub-swarm* di inisialisasi dan dibahas secara rinci pada sub-bab 4.5.1.1.



Gambar 4.6 Diagram alir inisialisasi partikel awal

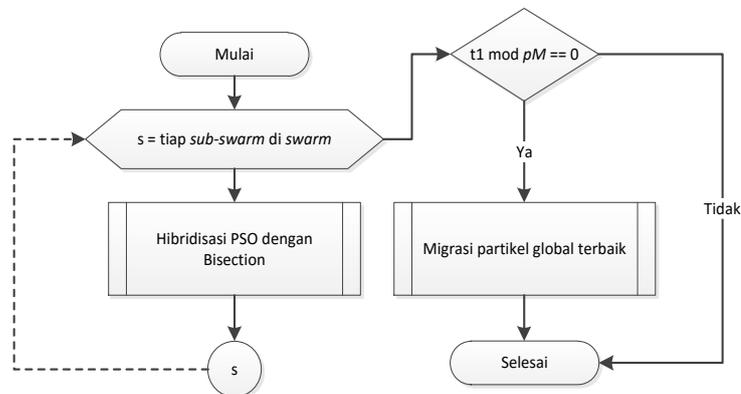
4.5.1.1 Inisialisasi Partikel

Inisialisasi partikel digunakan untuk membangkitkan partikel atau kandidat solusi awal yang akan di optimasi oleh PSO. Partikel sebanyak N (*swarm size*) dibangkitkan secara random antara 0 sampai 100 dalam bilangan riil positif. Pembangkitan partikel tersebut sangat memungkinkan tidak akan memenuhi *hard constraint*. Sehingga perbaikan partikel diperlukan dengan menggunakan Persamaan 4.31. Secara rinci proses inisialisasi partikel dapat dilihat pada diagram alir yang tergambar pada Gambar 4.6.

Pertama-tama nilai riil antara $[0,100]$ dibangkitkan secara random dan diperbaiki dengan Persamaan 4.31. Lalu partikel tersebut dievaluasi dengan fungsi *fitness* yang akan mengukur kualitas partikel dan nilai *fitness* di partikel tersebut akan disimpan dalam memori partikel. Sehingga, posisi terbaik personal atau posisi kognitif (*pBest*) menjadi partikel itu sendiri dikarenakan partikel belum mengalami proses pergerakan sehingga pengalaman terbaik tiap-tiap partikel untuk saat ini adalah posisi itu sendiri. Selanjutnya setelah semua partikel dibangkitkan, posisi terbaik global atau posisi sosial (*gBest*) akan dibangkitkan berdasarkan partikel terbaik di *swarm* tersebut yang dilihat dari nilai *fitness* tertinggi yang diperoleh dari proses evaluasi fungsi *fitness*.

4.5.2 Pergerakan Fase Migrasi

Pergerakan fase migrasi merupakan pergerakan partikel yang melibatkan pergerakan hibridisasi PSO dengan biseksi dan secara periodik partikel global terbaik akan dipindah posisikan ke *sub-swarm* tetangganya. Diagram alir pergerakan partikel pada fase migrasi dapat dilihat pada Gambar 4.7.

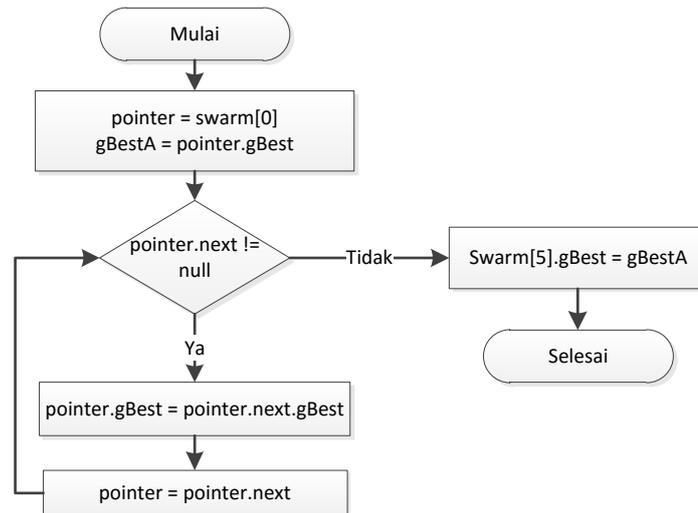


Gambar 4.7 Diagram alir fase migrasi

Tiap iterasi pada fase migrasi, nilai kecepatan dan posisi pada dimensi ke- d dan partikel ke- i yang berada pada *sub-swarm* s berubah menggunakan pergerakan partikel PSO yang ditingkatkan dengan metode biseksi pada tiap *sub-swarm*. Strategi pergerakan PSO dengan biseksi dan standar dibahas pada sub-bab 4.5.4. Selanjutnya setelah iterasi dijalankan sebanyak 5.000 iterasi (dinyatakan sebagai $t1 \bmod pM == 0$) secara periodik, maka migrasi pada *gBest* dilakukan untuk menjaga atau meningkatkan keberagaman tiap-tiap *sub-swarm*. Proses migrasi dibahas pada sub-bab 4.5.2.1.

4.5.2.1 Migrasi Partikel Global Terbaik

Migrasi partikel global terbaik dilakukan secara periodik dengan memanfaatkan struktur data *linked list*. Proses migrasi tersebut dapat dilihat pada Gambar 4.8.

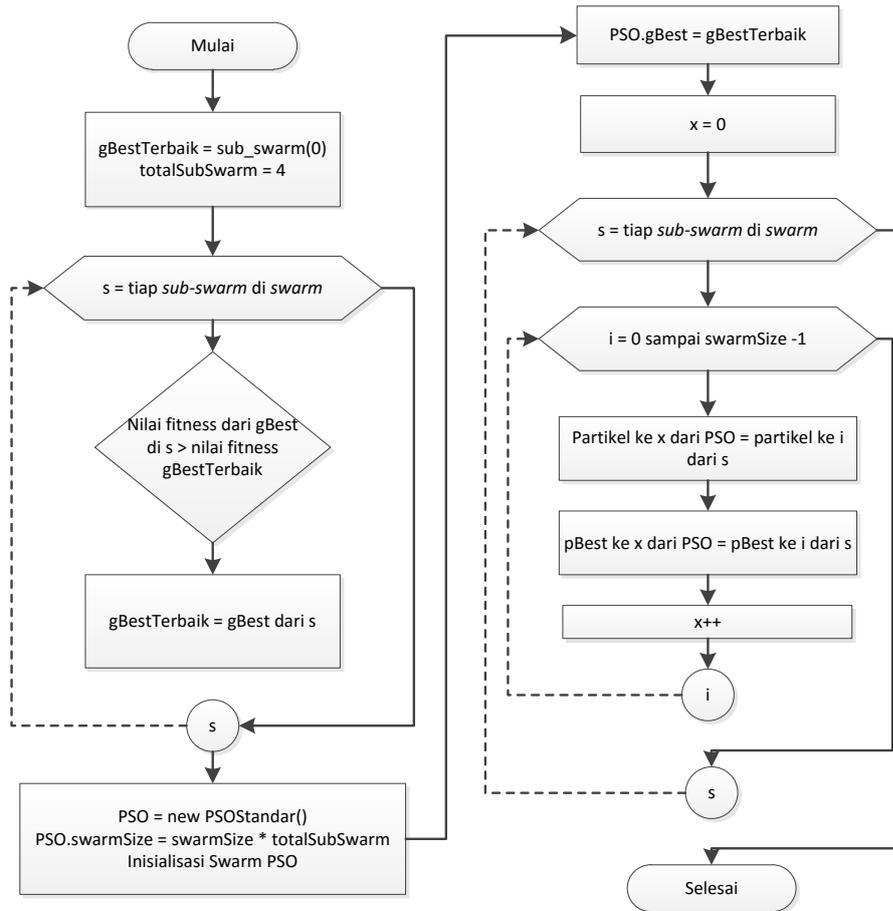


Gambar 4.8 Diagram alir migrasi partikel terbaik

Setiap *sub-swarm* memiliki variabel *next* yang merujuk pada *sub-swarm* tetangganya. Misal *sub-swarm* A (*swarm[0]*) dapat langsung merujuk ke *sub-swarm* B (*swarm[1]*) melalui variabel *next* begitu seterusnya sampai pada *sub-swarm* F yang variabel *next* merupakan null. Sehingga *sub-swarm* berupa untaian yang sekaligus merepresentasikan struktur data *linked list*. Untuk dapat memigrasikan *gBest* dengan perpindahan sirkular, pertama-tama *gBest* dari *sub-swarm* A ditampung terlebih dahulu pada variabel *gBestA*. Lalu *pointer* ditentukan merujuk pada *sub-swarm* A. Lalu selama *pointer* tersebut tidak sama dengan null, *gBest* pada *pointer* tersebut diganti menjadi *gBest* pada *sub-swarm* berikutnya (*pointer.next*). Lalu *pointer* merupakan *sub-swarm* berikutnya, begitu seterusnya sampai *pointer* null yang menunjukkan tidak ada lagi untaian *sub-swarm*. Karena *gBest* pada *sub-swarm* D belum merupakan *gBest* yang dimigrasikan, maka *gBest* pada *sub-swarm* D merupakan *gBestA* yang telah disimpan sebelumnya. Dengan proses seperti ini, *gBest* bermigrasi secara sirkular ke tetangga-nya.

4.5.3 Persiapan Fase Agregasi

Persiapan fase agregasi merupakan suatu prosedur yang diterapkan sebelum proses pergerakan pada fase agregasi dilakukan. Hal ini diperlukan karena masing-masing *sub-swarm* menyimpan informasi sendiri sehingga informasi pada tiap-tiap *sub-swarm* perlu digabungkan dengan mempertimbangkan partikel global terbaik dan menggabungkan seluruh partikel di tiap-tiap *sub-swarm* menjadi satu *swarm*. Diagram alir persiapan fase agregasi dapat dilihat pada Gambar 4.9.



Gambar 4.9 Persiapan fase agregasi

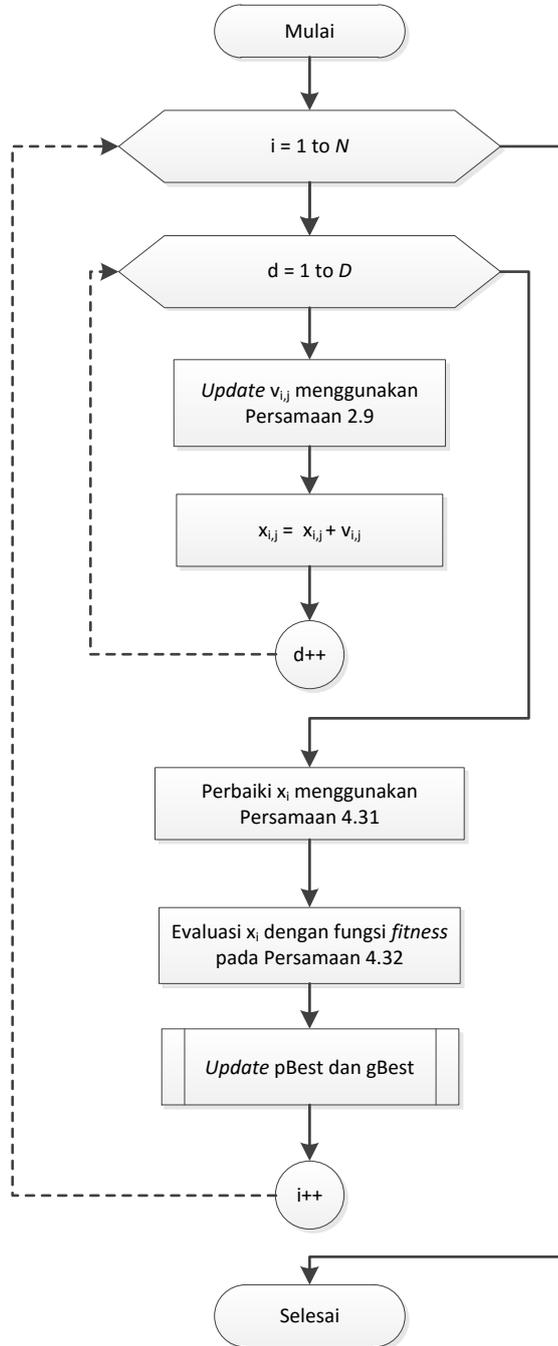
Mula-mula $gBest$ terbaik dari semua *sub-swarm* ditetapkan pada *sub-swarm* ke-0 atau A dan total *sub-swarm* sebesar 4. Selanjutnya, $gBest$ dengan nilai *fitness* tertinggi dicari dari tiap-tiap *sub-swarm* yang disimpan pada $gBestTerbaik$ melalui iterasi pada tiap *sub-swarm*. Lalu PSO dengan pergerakan standar di instansiasi dengan *swarm size* sebanyak total seluruh *swarm size* yang ada pada seluruh *sub-swarm* dan partikel di inialisasi. Tiap-tiap partikel yang ada pada *sub-swarm* di simpan pada *swarm* yang baru beserta dengan posisi personal terbaik. *Swarm* dengan PSO standar inilah yang akan digunakan pada fase agregasi dengan menggunakan pergerakan PSO standar.

4.5.4 Strategi Pergerakan PSO

Strategi *sub-swarm* PSO dari model hibridisasi terdiri dari 2 yaitu, standar dan hibridisasi PSO dengan biseksi. Strategi standar merupakan strategi yang memanfaatkan pergerakan PSO standar menggunakan Persamaan 2.9 dan 2.10 yang melibatkan beban inersia dan komponen akselerasi. Strategi hibridisasi PSO dengan biseksi merupakan strategi yang memanfaatkan biseksi dalam pergerakan partikel. Strategi tersebut dibahas secara rinci pada sub-bab 4.5.4.1.

Strategi standar memanfaatkan proses pergerakan partikel menggunakan Persamaan 2.9 dan 2.10. $x_{i,j}$ menunjukkan posisi pada partikel ke- i dan dimensi ke-

j yang merepresentasikan komposisi bahan pakan. $v_{i,j}$ merupakan kecepatan yang merubah posisi $x_{i,j}$. Kecepatan diperbarui setiap iterasinya yang terinflueni oleh beban inersia, komponen kognitif, dan komponen sosial berdasarkan Persamaan 2.9. Secara rinci proses pergerakan PSO standar tergambar pada Gambar 4.10. N merupakan *swarm size* dan D merupakan jumlah dimensi.

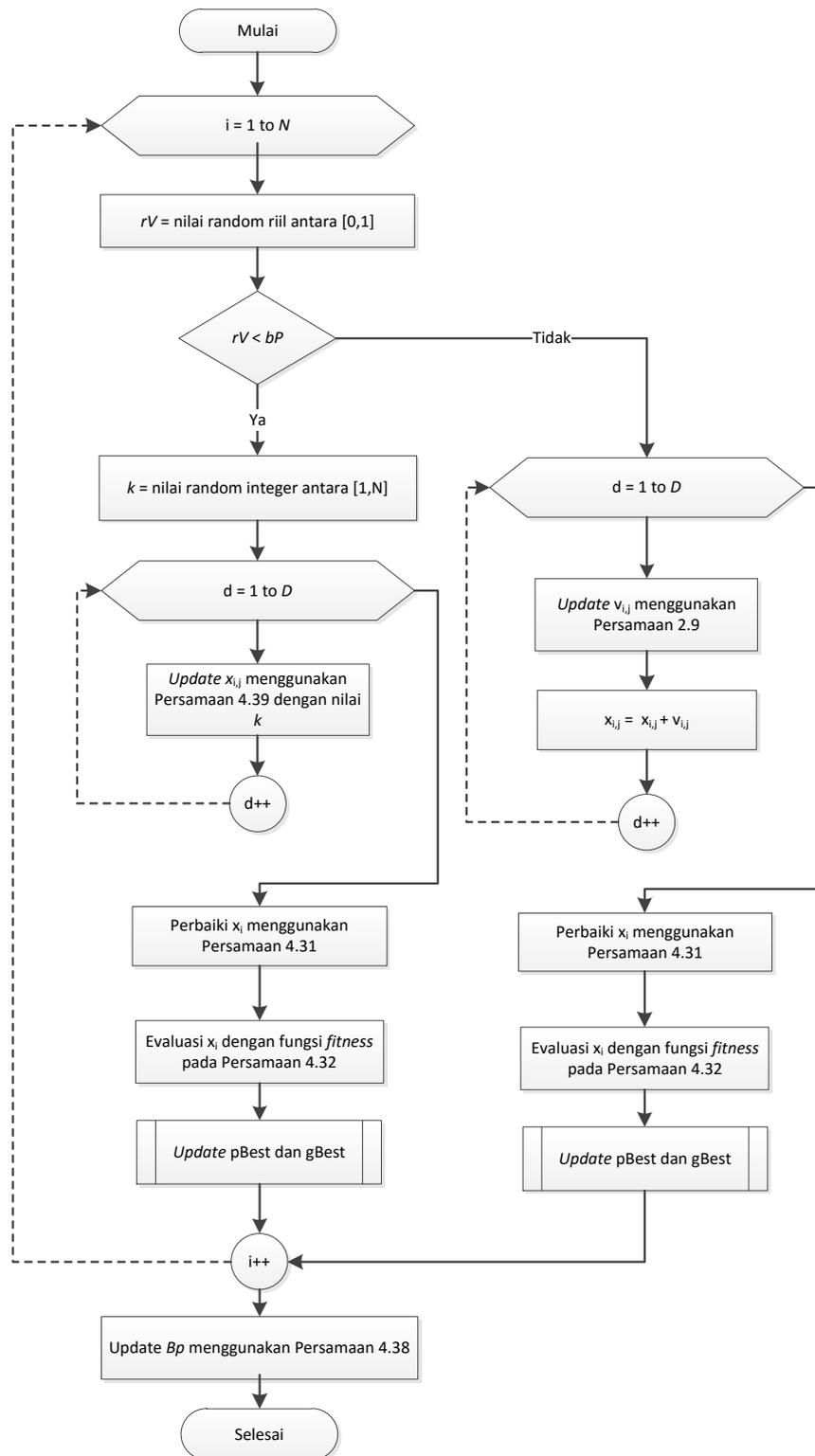


Gambar 4.10 Diagram alir strategi pergerakan PSO standar

4.5.4.1 Hibridisasi PSO dengan Biseksi

Selama pergerakan partikel, kecepatan partikel di PSO hanya dipengaruhi pengalaman posisi terbaik kognitif dan posisi terbaik global. Jika posisi yang lebih

baik terdapat diantara dua partikel, maka akan sangat memungkinkan untuk meningkatkan kinerja PSO. Oleh sebab itu, biseksi digunakan sebagai metode untuk pertukaran informasi antar dua partikel.

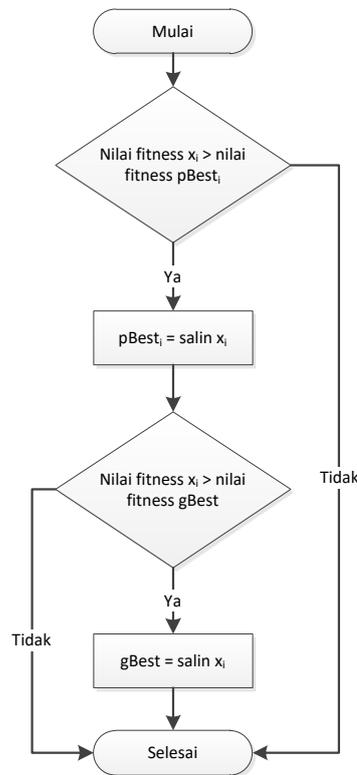


Gambar 4.11 Diagram alir strategi hibridisasi PSO dengan biseksi

$$Bp = (maxBp - minBp) \times \frac{maxIterasi-t}{maxIterasi} + minBp \quad (4.38)$$

$$x_{i,j}(t + 1) = \frac{x_{i,j}(t) + pbest_k}{2} \quad (4.39)$$

Metode biseksi akan dijalankan jika peluang biseksi (Bp) terpenuhi. Bp akan berkurang secara linear tiap iterasi menggunakan Persamaan 4.38. $maxIterasi$ merupakan maksimum iterasi dari fase migrasi dan t merupakan iterasi pada fase migrasi. $maxBp$ merupakan maksimum peluang dari Bp dan $minBp$ merupakan minimum peluang dari Bp . Nilai random riil akan dibangkitkan antara $[0,1]$ dan jika memenuhi Bp maka Persamaan 4.39 akan dijalankan. K merupakan nilai random integer antara $[0,N]$ yang dipastikan bahwa $i \neq k$.

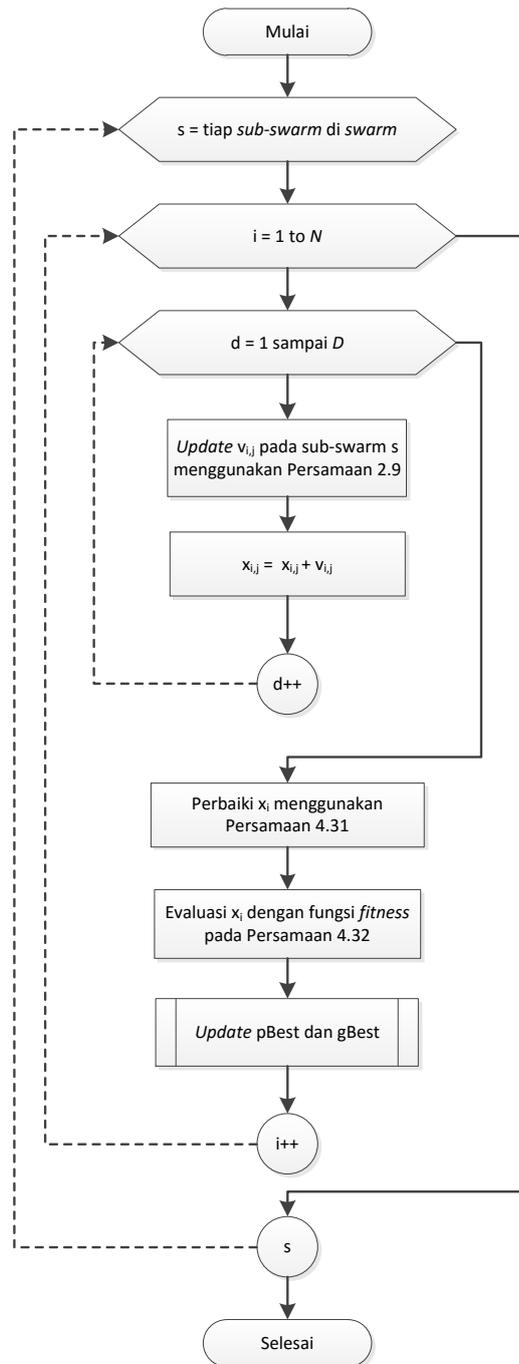


Gambar 4.12 Diagram alir *update pBest dan gBest*

Secara rinci proses pergerakan partikel menggunakan peluang biseksi tergambar pada Gambar 4.11. nilai rV dibangkitkan secara random dalam bilangan riil antara $[0,1]$. Jika rV kurang dari Bp maka pergerakan partikel berdasarkan biseksi dijalankan yang menggunakan persamaan 4.27 dengan membangkitkan nilai random integer k terlebih dahulu antara $[1,N]$. Sebaliknya jika rV kurang dari Bp tidak terpenuhi maka pergerakan PSO standar dilakukan menggunakan Persamaan 2.9 dan 2.10. Selanjutnya peluang Bp diturunkan secara linear diturunkan berdasarkan Persamaan 4.26.

4.5.4.2 Update $pBest$ dan $gBest$

Proses update $pBest$ dan $gBest$ tergambar pada Gambar 4.12. Pertama-tama nilai $fitness$ partikel dibandingkan dengan nilai $fitness pBest$. Jika menunjukkan nilai $fitness$ yang lebih baik maka $pBest$ akan diperbarui sesuai dengan partikel tersebut. Selanjutnya jika nilai $fitness$ partikel dibandingkan dengan nilai $fitness gBest$. Jika menunjukkan adanya perbaikan maka $gBest$ diperbarui terhadap partikel tersebut.



Gambar 4.13 Diagram alir pergerakan fase agregasi

4.5.5 Pergerakan Fase Agregasi

Fase agregasi dijalankan setelah 80% dari total iterasi selesai dijalankan. Pada fase ini semua *sub-swarm* telah digabung menjadi satu *swarm*. Pergerakan partikel pada fase migrasi sama dengan pergerakan PSO standar yang menggunakan Persamaan 2.9 dan 2.10 dalam perubahan kecepatan dan posisi. Diagram alir pergerakan tersebut secara rinci tergambar pada Gambar 4.13.

4.6 Skenario Pengujian

Skenario pengujian dirancang untuk mengetahui seberapa besar pengaruh komponen-komponen hibridisasi PSO seperti *sub-swarm* dalam menghasilkan solusi optimal. Selain itu parameter-parameter PSO seperti *swarm size*, jumlah iterasi, koefisien akselerasi, dan *inertia weight* akan diuji dengan mencoba satu-persatu nilai pada parameter tersebut. Semakin tinggi nilai *fitness* yang dihasilkan menunjukkan semakin optimal solusi yang dihasilkan. Karena PSO termasuk optimasi stokastik yang dapat menghasilkan solusi yang fluktuatif setiap kali dijalankan maka pengujian dilakukan sebanyak 10 kali dan rata-rata dari hasil tersebut dianalisis. Berikut dibahas spesifikasi pengujian, bahan pakan uji, dan skenario pengujian.

4.6.1 Spesifikasi Pengujian

Sub-bab ini menjelaskan spesifikasi pengujian yang akan dilakukan meliputi spesifikasi perangkat keras dan spesifikasi perangkat lunak yang di gunakan dalam menguji model hibridisasi PSO yang diajukan. Spesifikasi perangkat keras dan lunak masing-masing dapat dilihat pada Tabel 5.1 dan 5.2.

Tabel 4.1 Spesifikasi perangkat keras

<i>Processor</i>	Intel(R) Core(TM) i5-7200U CPU @2.50GHz 2.71GHz
<i>Memory (RAM)</i>	8,00 GB

Tabel 4.2 Spesifikasi perangkat lunak

<i>Operating System</i>	Windows 10 64-bit
<i>Programming Language</i>	Scala 2.12.2

4.6.2 Bahan Pakan Uji

Daftar bahan pakan yang berisi nilai nutrisi dan harga dapat dilihat pada Lampiran A. Kombinasi bahan pakan uji dengan jumlah kombinasi berbeda-beda dapat dilihat pada Tabel 4.1.

Tabel 4.3 Daftar bahan pakan uji

No.	Bahan Pakan Uji	Bahan Pakan
1	A11	0,1,2,3,4,9,10,11,12,14,26

2	A12	0,2,4,6,10,13,14,20,21,22,26,27
3	A13	1,2,3,4,5,8,17,18,21,25,26,27,28
4	A14	0,1,2,3,4,5,10,15,16,18,21,24,25,26
5	A15	0,1,2,3,7,8,9,10,19,25,22,23,24,26,27
6	B11	1,2,3,4,5,9,10,16,17,18,26
7	B12	0,1,3,6,10,13,14,16,20,22,26,27
8	B13	0,1,2,3,5,8,10,14,17,23,25,27,28
9	B14	0,1,2,6,7,10,11,14,16,21,22,24,25,28
10	B15	0,1,2,5,6,7,9,10,13,14,20,23,25,26,27

A11 menunjukkan bahan pakan uji yang memiliki jumlah bahan pakan yang berbeda sebanyak 11 dan begitu seterusnya. Sedangkan B11 menunjukkan jumlah bahan uji yang sama seperti A11 yaitu 11, namun memiliki kombinasi bahan pakan yang berbeda. Daftar bahan pakan uji A11-A15 digunakan dalam pengujian komponen hibridisasi, jumlah *sub-swarm*, maksimum probabilitas biseksi, *swarm size*, jumlah iterasi, koefisien akselerasi, dan *inertia weight*. Sedangkan pada skenario pengujian komparasi digunakan bahan uji yang berbeda namun dengan jumlah kombinasi yang sama yaitu menggunakan bahan uji B11-B15 (Silakan lihat lampiran A untuk melihat lebih rinci bahan pakan yang digunakan dalam formula tersebut).

4.6.3 Pengujian Mekanisme Hibridisasi

Pengujian mekanisme hibridisasi dirancang untuk mengetahui apakah komponen hibridisasi seperti pergerakan partikel menggunakan biseksi pada *sub-swarm* dan fase agregasi dapat memberikan solusi yang lebih baik. Mekanisme hibridisasi yang diusulkan terdiri dari 4 *sub-swarm* yang melibatkan probabilitas pergerakan biseksi. Setiap periode tertentu posisi global terbaik dimigrasikan secara sirkular ke *sub-swarm* tetangganya. Lalu setelah proses pergerakan tersebut selesai (fase migrasi), semua *sub-swarm* digabung menjadi satu *swarm* dengan posisi global terbaik berupa posisi global terbaik dari semua *sub-swarm* lalu pergerakan partikel menggunakan pergerakan PSO standar (fase agregasi).

Untuk menguji usulan hibridisasi tersebut, mekanisme hibridisasi dibagi menjadi 4 mekanisme. Mekanisme pertama adalah terdapat salah satu *sub-swarm* yang tidak melibatkan biseksi atau menggunakan PSO standar (Mekanisme A). Mekanisme kedua adalah semua *sub-swarm* menggunakan pergerakan partikel dengan probabilitas biseksi atau merupakan hibridisasi PSO yang diusulkan (Mekanisme B). Mekanisme ketiga adalah fase agregasi tidak dilibatkan pada hibridisasi PSO (Mekanisme C). Dan mekanisme terakhir adalah fase agregasi dilibatkan dengan menggunakan pergerakan probabilitas biseksi (Mekanisme D). Lebih rinci mekanisme hibridisasi dijabarkan pada Tabel 4.2.

Tabel 4.4 Mekanisme hibridisasi

Mekanisme Hibridisasi	Keterangan
A	PSO standar, PSO Biseksi dengan MaxBp = 0,2, 0,4, dan 0,6
B	PSO Biseksi dengan MaxBp = 0,2, 0,4, 0,6, dan 0,8
C	PSO Biseksi dengan MaxBp = 0,2, 0,4, 0,6, dan 0,8 tanpa fase agregasi
D	PSO Biseksi dengan MaxBp = 0,2, 0,4, 0,6, dan 0,8 dengan fase agregasi melibatkan pergerakan biseksi.

4.6.4 Pengujian Maksimum Probabilitas Biseksi (MaxBp)

Setiap *sub-swarm* partikel akan bergerak menggunakan Persamaan 4.39 jika probabilitas terpenuhi. Probabilitas akan menurun seiring dengan bertambahnya jumlah iterasi dan nilai awal pada probabilitas tersebut (MaxBp) seperti pada Persamaan 4.38 perlu ditemukan nilai yang optimum. Sehingga parameter MaxBp akan dilakukan *tuning* terhadap 5 bahan uji yang berbeda (A11 – A15).

Nilai parameter MaxBp disesuaikan mulai dari 0,1 sampai dengan 1,0 untuk semua *sub-swarm*. Nilai MaxBp yang dapat memberikan rata-rata *fitness* terbaik dan nilai MaxBp di atasnya tidak menunjukkan adanya perubahan atau bahkan menurunkan rata-rata *fitness* dipilih sebagai parameter yang optimum. Hasil dari masing-masing MaxBp yang optimum untuk tiap-tiap bahan uji akan di bandingkan dengan gabungan dari nilai optimum MaxBp yang ditemukan untuk tiap-tiap *sub-swarm*. Pengujian tersebut dilakukan untuk mengetahui apakah gabungan nilai optimum MaxBp dapat memberikan solusi yang baik.

4.6.5 Pengujian Jumlah Sub-Swarm

Pengujian jumlah *sub-swarm* dirancang untuk dapat mengetahui pengaruh banyak-nya *sub-swarm* terhadap hasil rata-rata *fitness* yang dihasilkan. Pengujian ini dirancang dengan menguji jumlah *sub-swarm* terhadap rata-rata nilai *fitness* yang dihasilkan. Pengujian ini menggunakan 5 bahan uji (A11- A15) yang dapat dilihat pada Tabel 4.1.

Swarm size untuk keseluruhan *sub-swarm* ditentukan sama walaupun jumlah *sub-swarm* berbeda. Sebagai contoh, pada pengujian *swarm size* sebesar 60 untuk 2 *sub-swarm* maka tiap-tiap *sub-swarm* memiliki *swarm size* sebesar 30. Sedangkan untuk 4 *sub-swarm*, tiap-tiap *sub-swarm* memiliki *swarm size* sebesar 15. Sehingga total semua *swarm size* pada tiap-tiap *sub-swarm* dipastikan sama dengan 60.

Mula-mula *swarm size* ditentukan sebesar 60 dan meningkat sampai 300 *swarm size* dengan tingkatan sebesar 60. Nilai 60 dipilih karena nilai tersebut dapat mudah dibagi oleh 1 sampai 6, sehingga proses pembagian nilai *swarm size*

akan lebih mudah yang menyebabkan jumlah swarm size pada tiap-tiap *sub-swarm* berjumlah sama. Pengujian ini ditunjukkan untuk melihat adanya pengaruh jumlah *sub-swarm* dengan rata-rata *fitness* yang dihasilkan.

4.6.6 Pengujian Swarm Size

Pengujian *swarm size* dirancang untuk mengetahui seberapa besar *swarm size* yang diperlukan untuk dapat menghasilkan formula yang optimal dengan jumlah kombinasi bahan pakan yang berbeda-beda. Bahan pakan uji yang digunakan adalah mulai dari A11 sampai A15. *Swarm size* mulai dari 5 dengan kenaikan sebesar 5 akan disesuaikan dengan rata-rata nilai *fitness* yang dihasilkan. Penyesuaian *swarm size* akan dihentikan jika rata-rata nilai *fitness* sudah tidak menunjukkan perubahan yang signifikan dan *swarm size* yang baik ditentukan dari hasil uji yang diperoleh. Nilai *swarm size* yang optimum akan ditentukan dari kelima bahan uji yang berbeda.

4.6.7 Pengujian Jumlah Iterasi

Pengujian iterasi dirancang untuk dapat mengetahui jumlah iterasi yang diperlukan untuk menghasilkan solusi yang optimal. Pengujian ini juga akan dilakukan pada jumlah kombinasi bahan pakan yang berbeda-beda untuk mengetahui pengaruh jumlah iterasi yang diperlukan terhadap jumlah bahan pakan. Bahan pakan uji yang digunakan adalah mulai dari A11 sampai A15 yang dapat dilihat pada Tabel 4.1. Jumlah iterasi akan disesuaikan mulai dari 1,000 yang akan dinaikan sebesar 1,000 setiap penyesuaian terhadap rata-rata *fitness*. Penyesuaian jumlah iterasi akan dihentikan jika rata-rata nilai *fitness* sudah tidak menunjukkan perubahan yang signifikan dan jumlah iterasi yang baik ditentukan dari hasil uji yang diperoleh.

4.6.8 Pengujian Parameter PSO

Pengujian parameter PSO dirancang untuk dapat mengetahui nilai parameter yang diperlukan untuk menghasilkan solusi yang optimal. Pengujian ini juga akan dilakukan pada jumlah kombinasi bahan pakan yang berbeda-beda untuk mengetahui pengaruh nilai parameter yang diperlukan terhadap kombinasi bahan pakan. Bahan pakan uji yang digunakan adalah mulai dari A11 sampai A15 yang dapat dilihat pada Tabel 4.1 dengan melibatkan nilai hasil pengujian *swarm size* dan iterasi sebagai *swarm size* dan iterasi yang optimum.

Mula-mula nilai parameter koefisien kognitif (c_1) dan koefisien sosial (c_2) akan disesuaikan melalui kombinasi mulai dari 0,1 sampai dengan 2,0 dengan kenaikan nilai sebanyak 0,1. Setelah nilai koefisien kognitif dan sosial yang optimum ditentukan, nilai *inertia weight* (w) akan disesuaikan mulai dari 0,1 sampai dengan 1,0 dengan kenaikan nilai sebanyak 0,1 dengan melibatkan nilai koefisien kognitif dan sosial yang optimum.

4.6.9 Pengujian Komparasi

Pengujian komparasi dirancang untuk mengetahui kualitas solusi hibridisasi PSO dibandingkan dengan metode komputasi evolusioner lain. Algoritme yang

akan dibandingkan adalah PSO, GA, dan Hibridisasi GA Adaptif dengan SA (AGASA). Parameter AGASA terdiri dari jumlah populasi sebesar 400, jumlah generasi sebesar 300, *crossover rate* sebesar 0,6, *mutation rate* sebesar 0,4, iterasi SA sebesar 10, laju penurunan suhu sebesar 0,75, dan jumlah generasi adaptif sebesar 3. Namun jumlah populasi dan jumlah generasi mengikuti total *swarm size* dan total iterasi yang baik yang dihasilkan dari skenario pengujian *swarm size* dan total iterasi MSPSO agar pengujian yang dilakukan adil karena menggunakan parameter yang sama.

Hasil pengujian komparasi yang dibandingkan antara lain rata-rata *fitness*, rata-rata *penalty*, rata-rata biaya, rata-rata waktu komputasi, dan rata-rata standar deviasi. Selain itu penambahan beban pada salah satu kendala juga diuji untuk mengetahui seberapa kekar MSPSO dalam beradaptasi dengan penambahan beban tersebut. Bahan pakan uji yang digunakan adalah B11 sampai B15.