

BAB 4 PERANCANGAN

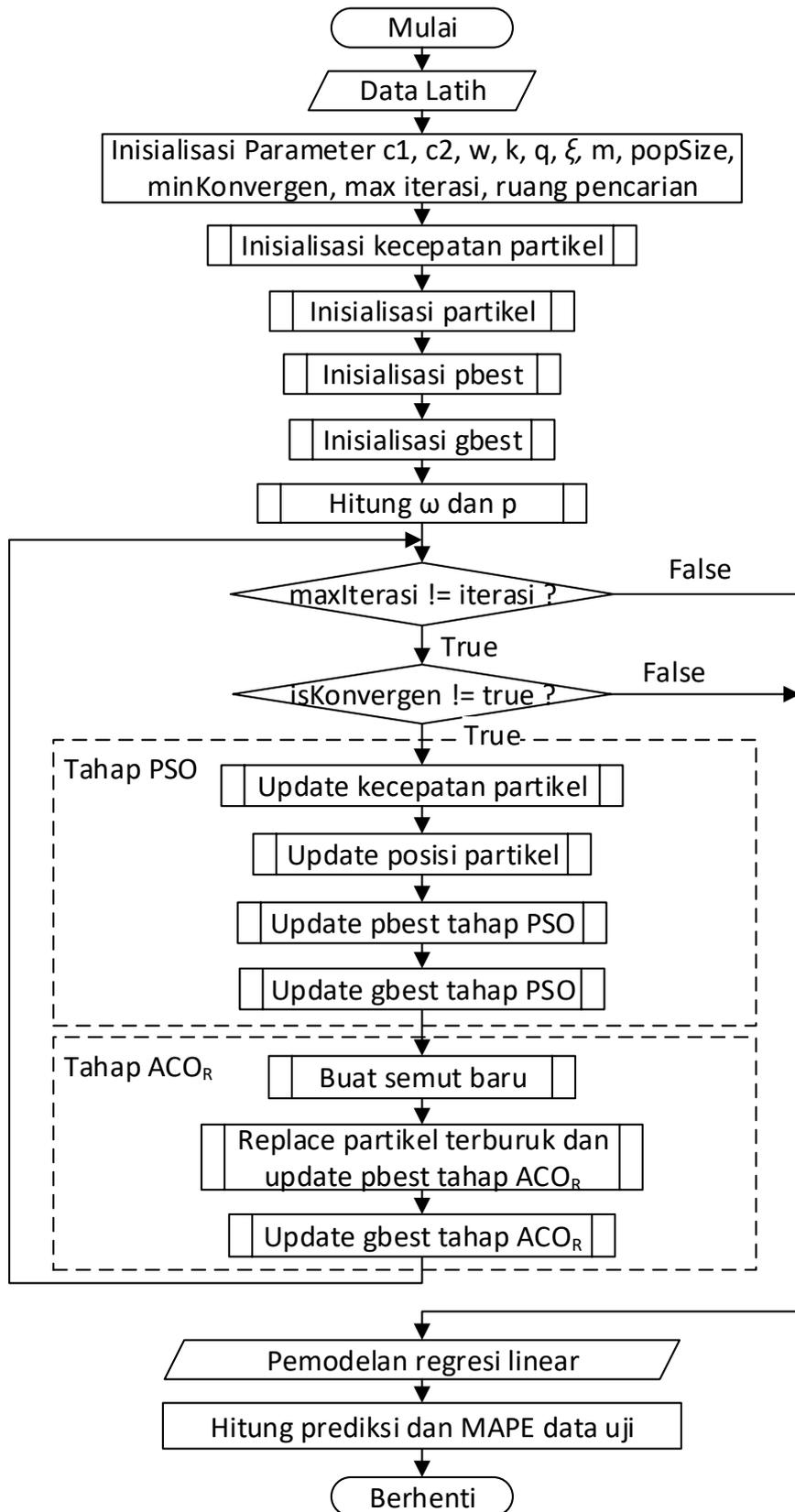
4.1 Perancangan Algoritma

4.1.1 Hybrid PSO dan ACO_R

Hibridasi yang dilakukan menggunakan pendekatan sekuensial seperti pada Gambar 4.1. PSO dan ACO_R saling berbagi set solusi yang disebut partikel oleh PSO dan solusi pada ACO_R, keduanya direpresentasikan dalam bentuk tabel partikel. Tahap inialisasi dilakukan dengan menggunakan cara seperti PSO sedangkan pada tahap iterasi dilakukan dua tahap yakni tahap pertama dilakukan *update* menggunakan PSO berupa proses *update* kecepatan partikel, *update* posisi partikel, *update* pbest tahap PSO dan gbest tahap PSO. Sedangkan tahap kedua menggunakan ACO_R berupa proses buat semut baru, *replace* partikel terburuk dan *update* pbest tahap ACO_R jika *fitness* semut baru lebih baik dari *fitness* partikel terburuk kemudian terakhir *update* gbest tahap ACO_R jika *fitness* semut baru lebih baik daripada hasil *update* gbest tahap PSO.

Berikut detail tahap-tahap prediksi konsumsi energi primer menggunakan hybrid PSO dan ACO_R.

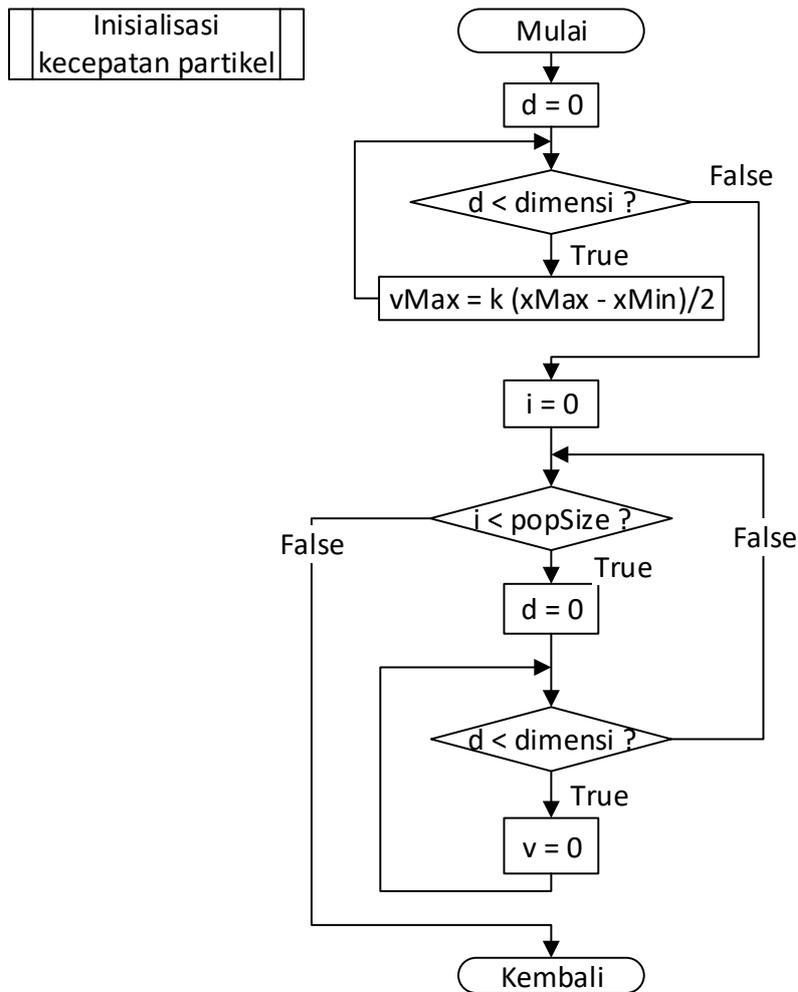
1. Meng-queri data latih dari tahun 1967 sampai 2006 sekaligus data uji mulai tahun 2007 sampai 2016 yang disimpan dalam database SQLite berupa data konsumsi energi primer, GNI, GDP, populasi, impor, dan ekspor Indonesia.
2. Menentukan parameter PSOACO_R berupa nilai c_1 , c_2 , w , k *velocity clamping*, q , ξ , m , $popSize$, min konvergen, dan max iterasi.
3. Inialisasi tabel partikel menggunakan Persamaan 2.2
4. Inialisasi kecepatan awal partikel dengan nilai sebesar 0 untuk semua partikel
5. Inialisasi pbest dengan menyalin semua nilai partikel awal beserta *fitness*-nya pada tabel partikel
6. Inialisasi gbest yakni pbest dengan *fitness* terkecil
7. Hitung nilai ω dengan Persamaan 2.11 dan peluang p dengan Persamaan 2.12
8. *Update* kecepatan partikel dengan Persamaan 2.3
9. *Update* posisi partikel menggunakan Persamaan 2.4
10. *Update* pbest tahap PSO menggunakan Persamaan 2.8 dengan membandingkan pbest iterasi sebelumnya dengan partikel baru. Jika *fitness* partikel baru ke- i lebih baik (dalam penelitian ini *fitness* yang lebih kecil) dari pbest iterasi ke- i iterasi sebelumnya maka *update* pbest tahap PSO berupa set solusi yang dibawa oleh partikel baru, begitu juga sebaliknya.
11. *Update* gbest tahap PSO (Persamaan 2.9. *Update* gbest dilakukan dengan mengganti gbest iterasi sebelumnya dengan pbest terbaik (terkecil) iterasi saat ini. Jika pbest terbaik memiliki *fitness* lebih baik daripada *fitness* gbest sebelumnya, maka ganti gbest iterasi sebelumnya dengan pbest terbaik.
12. Buat semut baru menggunakan Persamaan 2.13 sampai 2.15



Gambar 4.1 Flowchart pemodelan regresi-PSOACO_R untuk prediksi konsumsi energi primer Indonesia

13. *Replace* partikel terburuk (Persamaan 2.17) dan *update* pbest tahap ACO_R (Persamaan 2.18), jika *fitness* semut baru lebih baik daripada *fitness* terburuk partikel pada tabel partikel. *Replace* partikel terburuk berupa penggantian partikel terburuk dengan set solusi semut baru yakni jika *fitness* semut baru lebih baik daripada *fitness* partikel terburuk. Jika kondisi tersebut terpenuhi, maka lakukan juga *update* pbest tahap ACO_R menggunakan Persamaan 2.18 hanya pada pbest ke-*i* yang sejajar dengan partikel terburuk yang telah diganti. Saat *fitness* semut baru tidak lebih baik daripada *fitness* partikel terburuk maka lompat proses ini.
14. *Update* gbest tahap ACO_R seperti pada Persamaan 2.19 yakni jika *fitness* semut baru lebih baik daripada gbest tahap PSO maka lakukan *update* gbest.
15. Menghitung prediksi konsumsi energi primer menggunakan Persamaan 2.20 dan hitung MAPE data uji dengan Persamaan 2.22. Agar lebih memahami urutan prediksi konsumsi energi primer Indonesia dengan $PSOACO_R$ dapat dilihat pada flowchart Gambar 4.1.

4.1.2 Inisialisasi Kecepatan Partikel



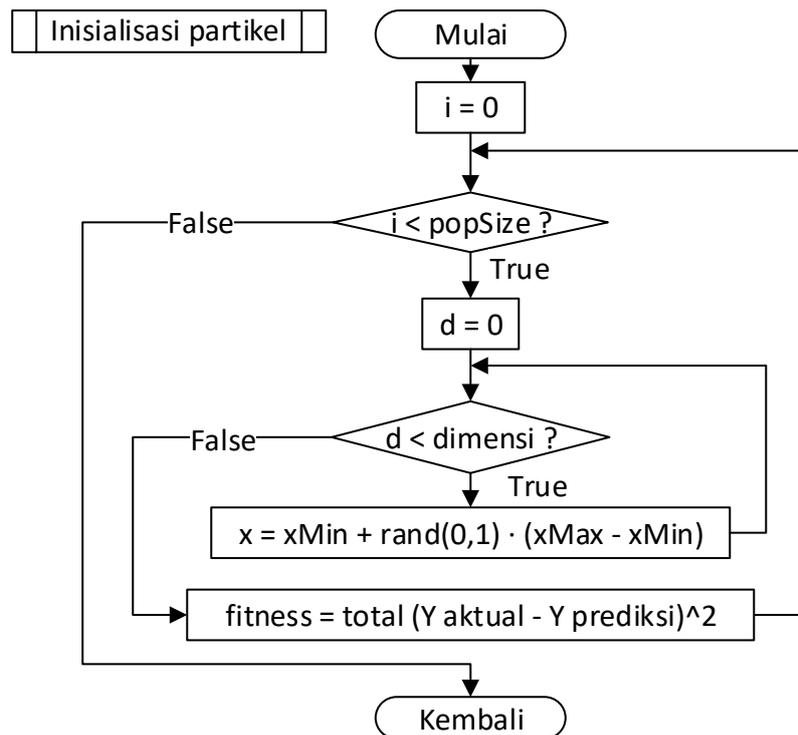
Gambar 4.2 Flowchart inisialisasi kecepatan partikel

Proses inialisasi kecepatan partikel dengan cara menentukan kecepatan awal masing-masing dimensi dalam setiap partikel berupa nilai kecepatan awal yang sama, yakni sebesar 0. Proses menentukan $-v_{max}$ dan v_{max} dengan Persamaan 2.4 diperlukan untuk *velocity clamping* saat *update* kecepatan partikel nanti, agar setiap partikel yang akan di-*update* kecepatannya tidak mengalami *velocity explosion*, akibatnya partikel mudah keluar dari ruang pencarian yang ditentukan. Sesuai penjelasan tersebut maka flowchart inialisasi kecepatan partikel dapat dilihat pada Gambar 4.2.

4.1.3 Inialisasi Partikel

PSO dan ACO_R saling berbagi menggunakan solusi yang sama pada tabel partikel. Pada PSO tabel set solusi disebut sebagai tabel partikel, sedangkan pada ACO_R disebut dengan tabel *archive*. Penjelasan tahapan untuk inialisasi tabel Partikel seperti pada Gambar 4.3 yakni:

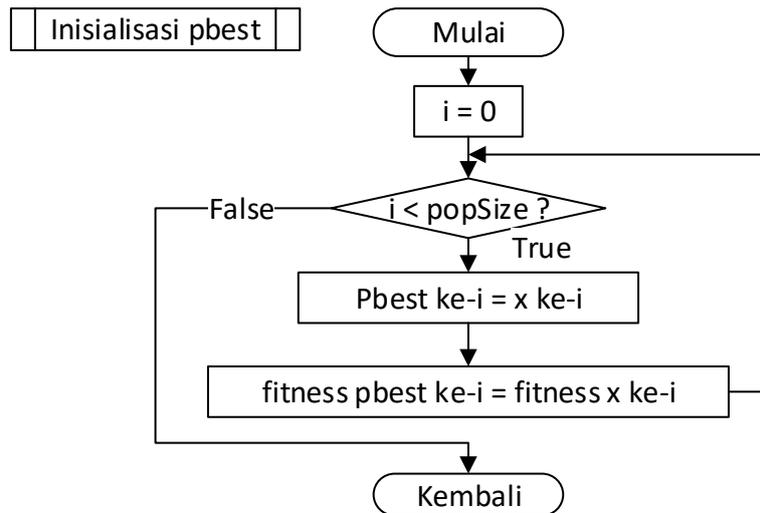
1. Buat partikel baru secara random menggunakan Persamaan 2.2. Random yang digunakan memiliki distribusi uniform antara 0 sampai 1.
2. Hitung *fitness* setiap partikel menggunakan Persamaan 2.21



Gambar 4.3 Flowchart inialisasi partikel

4.1.4 Inialisasi Pbest

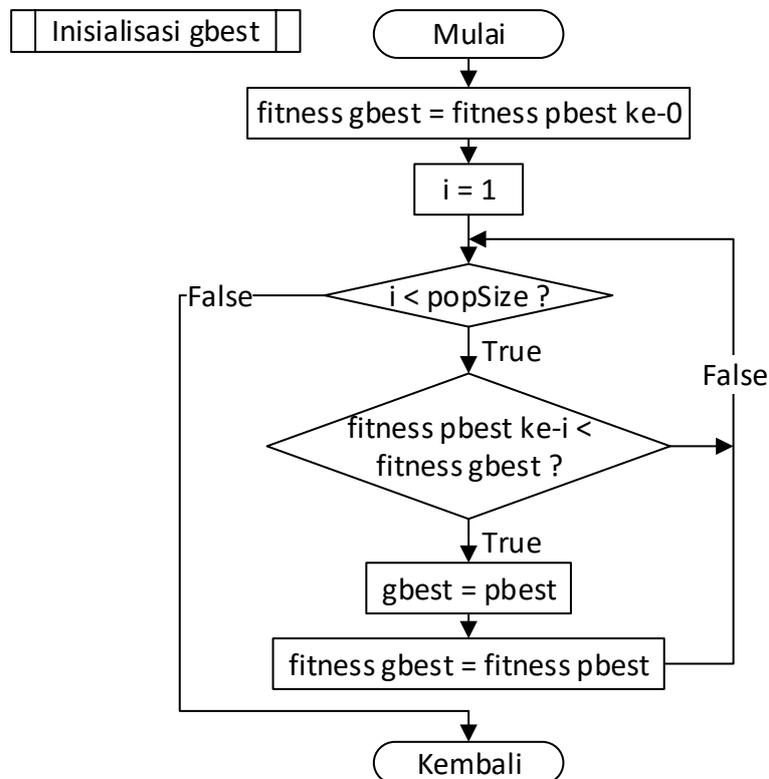
Pada proses inialisasi pbest, nilai setiap elemen didalamnya sama dengan nilai partikel (tabel partikel) itu sendiri beserta *fitness*-nya, sehingga untuk inialisasi pbest hanya perlu menyalin keseluruhan elemen partikel pada tabel partikel. Proses inialisasi pbest ditunjukkan pada flowchart pada Gambar 4.4.



Gambar 4.4 Flowchart inialisasi pbest

4.1.5 Inialisasi Gbest

Inialisasi gbest dicari dengan menemukan pbest dengan nilai *fitness* terbaik, dimana dalam permasalahan penelitian ini merupakan pbest dengan *fitness* terkecil. Pbest dengan *fitness* terkecil tersebut dicari dengan membandingkan *fitness* pbest ke-0 dengan *fitness* pbest ke-*i* lainnya. Proses inialisasi gbest dapat dilihat pada Gambar 4.5.



Gambar 4.5 Flowchart inialisasi gbest

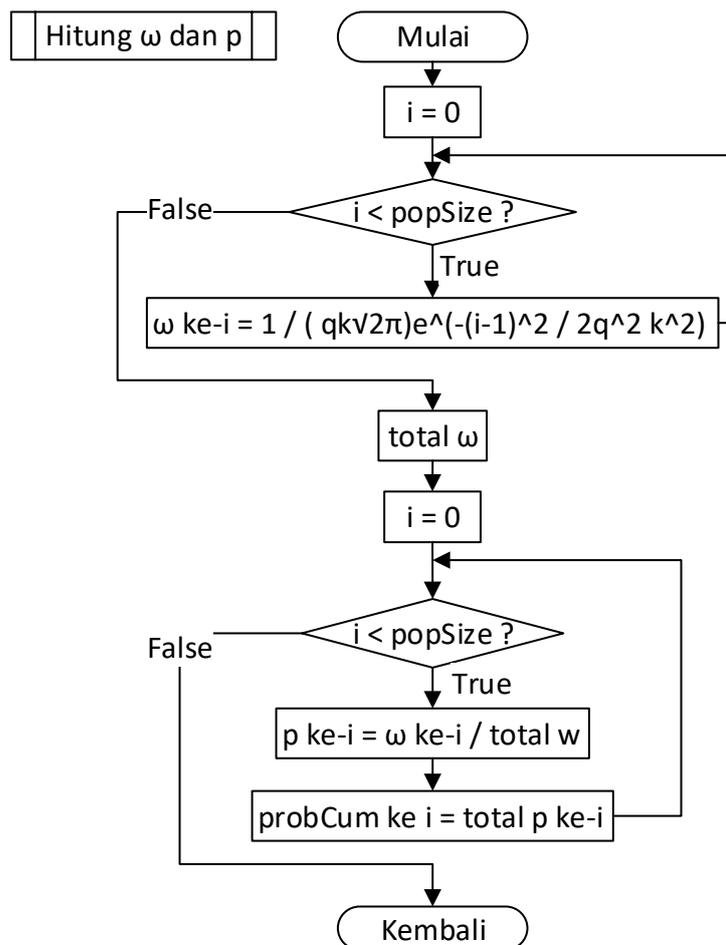
4.1.6 Hitung ω dan Peluang p

Proses menghitung nilai ω dan peluang p digunakan untuk proses seleksi menggunakan *roulette wheel* pada proses buat semut baru. Tahapan menghitung nilai ω dan peluang p yakni:

1. Menghitung bobot ω ke- i menggunakan Persamaan 2.11
2. Menghitung total bobot ω dengan menjumlahkan keseluruhan nilai bobot ω dari partikel pertama sampai terakhir
3. Menghitung nilai peluang p ke- i menggunakan Persamaan 2.12 yakni dengan cara membagi nilai ω ke- i dengan nilai total ω
4. Hitung peluang kumulatif ke- i dengan cara menjumlahkan semua peluang p mulai dari p pertama sampai p ke- i

Penentuan nilai ω dan p berguna untuk memilih partikel menggunakan *roulette wheel* sebagai s_g (Persamaan 2.13). Untuk permasalahan global minimum maka ω dengan besar nilai $\omega_1 \geq \omega_2 \geq \dots \geq \omega_i \geq \dots \geq \omega_k$ begitu pula dengan besar peluang p yakni $p_1 \geq p_2 \geq \dots \geq p_i \geq \dots \geq p_k$.

Berdasarkan tahapan tersebut dapat dibentuk *flowchart* pada Gambar 4.6.

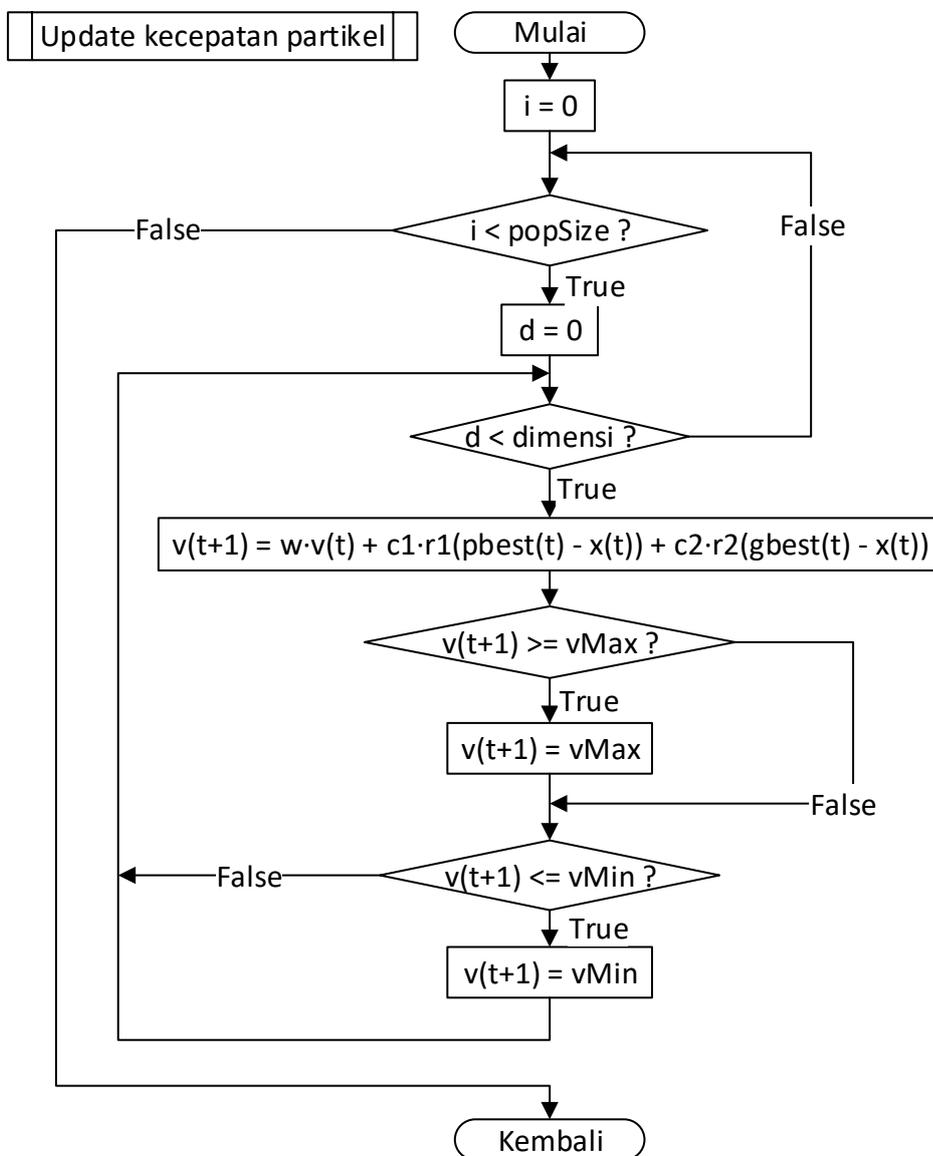


Gambar 4.6 Flowchart hitung nilai ω dan peluang p

4.1.7 Update Kecepatan Partikel

Update kecepatan partikel merupakan proses pertama setiap iterasinya dalam metode *hybrid* PSOACO_R. *Update* kecepatan berfungsi untuk mengubah posisi partikel agar semakin mendekati global minimum. Tahapan proses *update* kecepatan partikel sesuai Gambar 4.7 dapat dijelaskan sebagai berikut:

1. Untuk setiap kecepatan partikel ke-*i* dimensi ke-*d*, hitung kecepatan partikel ke-*i* dimensi ke-*d* menggunakan Persamaan 2.3
2. Lakukan pemeriksaan hasil *update* kecepatan untuk dilakukan pembatasan kecepatan atau *velocity clamping* agar setiap partikel yang akan di-*update* kecepatannya tidak mengalami *velocity explosion*. Jika kurang dari batas bawah maka ganti dengan nilai $-v_{max}$, jika melebihi dari batas atas maka *update* kecepatan diganti dengan dengan nilai v_{max} .

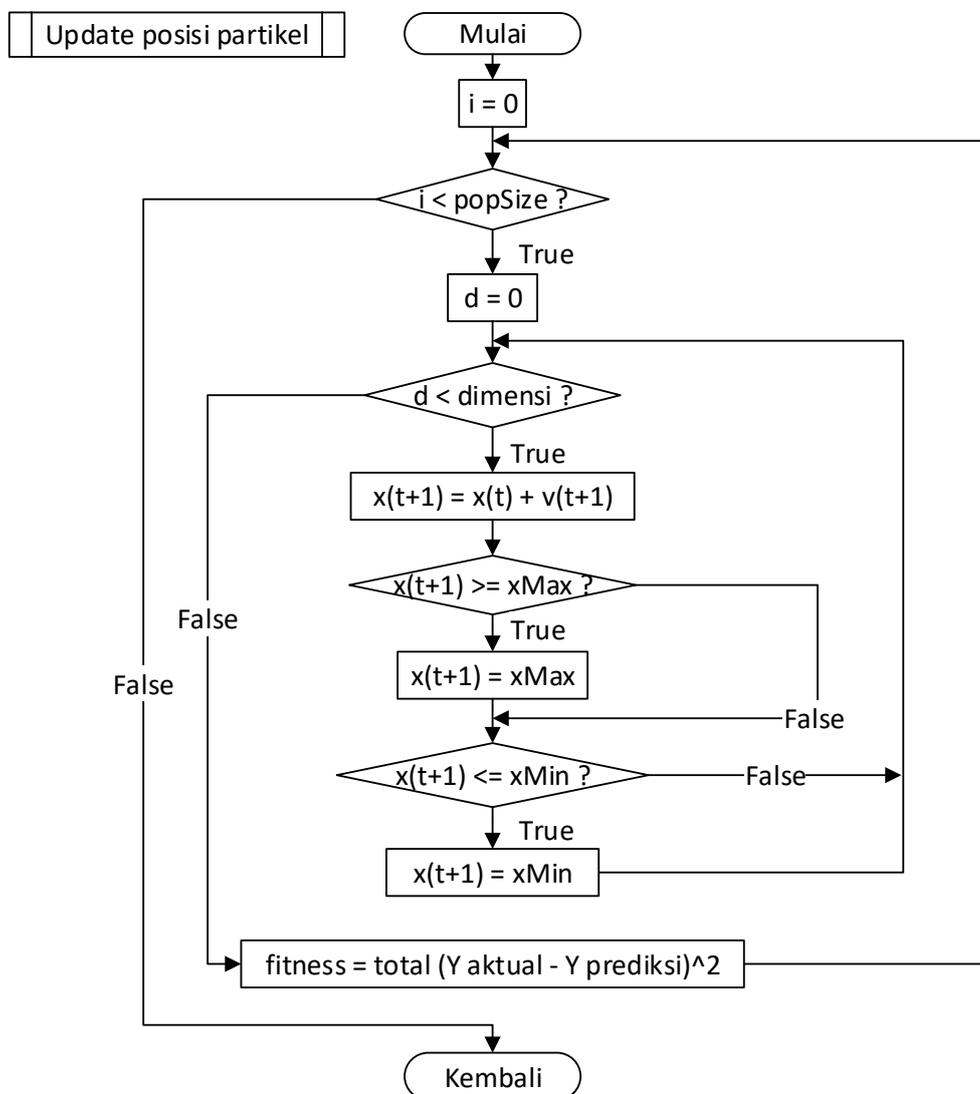


Gambar 4.7 Flowchart *update* kecepatan partikel

4.1.8 Update Posisi Partikel

Langkah-langkah proses *update* posisi partikel sesuai Gambar 4.8 dapat dijelaskan sebagai berikut:

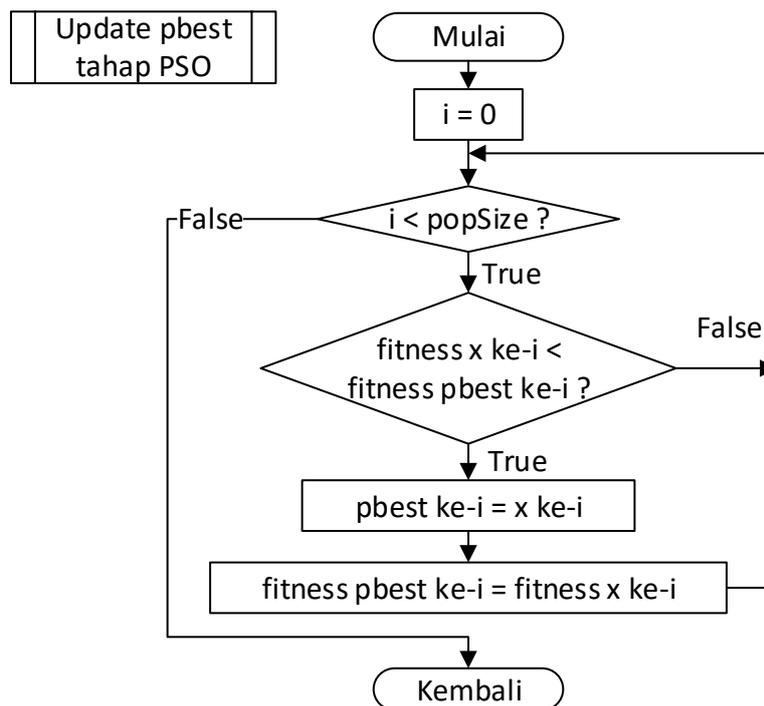
1. Hitung partikel baru ke- i dimensi ke- d menggunakan Persamaan 2.6
2. Meski sudah melakukan proses *velocity clamping*, posisi partikel masih bisa keluar dari ruang pencarian (*search space*) yang telah ditentukan, maka perlu dilakukan *limit* (pembatasan) posisi partikel menggunakan Persamaan 2.7. Lakukan pembatasan hasil partikel baru untuk setiap dimensi partikel ke- i , jika kurang dari batas bawah ruang pencarian maka ganti dengan nilai x_{min} , jika melebihi dari batas atas maka ganti posisi partikel dengan nilai x_{max} .



Gambar 4.8 Flowchart *update* posisi partikel

4.1.9 Update Pbest Tahap PSO

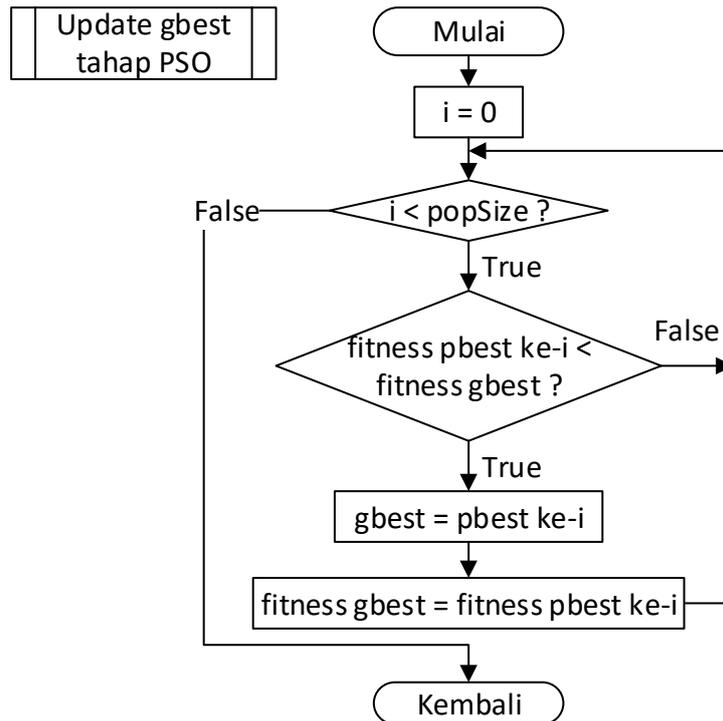
Update pbest tahap PSO sesuai dengan Persamaan 2.8 dilakukan dengan membandingkan semua nilai *fitness* pbest ke-*i* iterasi sebelumnya dengan *fitness* partikel baru ke-*i* dan dipilih yang terbaik. Jika *fitness* partikel baru ke-*i* lebih baik daripada *fitness* pbest ke-*i* iterasi sebelumnya maka ganti pbest yang sejajar dengan partikel baru tersebut, jika *fitness* partikel baru tidak lebih baik maka pbest tidak perlu diganti (tetap). Proses *update* pbest tahap PSO ditunjukkan pada flowchart pada Gambar 4.9.



Gambar 4.9 Flowchart *update* pbest tahap PSO

4.1.10 Update Gbest Tahap PSO

Terakhir pada siklus PSO berupa proses *update* gbest untuk menyimpan posisi terbaik yang dicapai oleh partikel. *Update* gbest dicari dengan melakukan perbandingan antara *fitness* gbest iterasi sebelumnya dengan *fitness* pbest terbaik, hal ini sesuai seperti pada Persamaan 2.9 tentang update gbest PSO. Jika *fitness* pbest terbaik lebih baik maka lakukan penggantian set solusi gbest, jika tidak maka set solusi gbest tetap. Proses *update global best* (gbest) tahap PSO ditunjukkan pada flowchart pada Gambar 4.10.

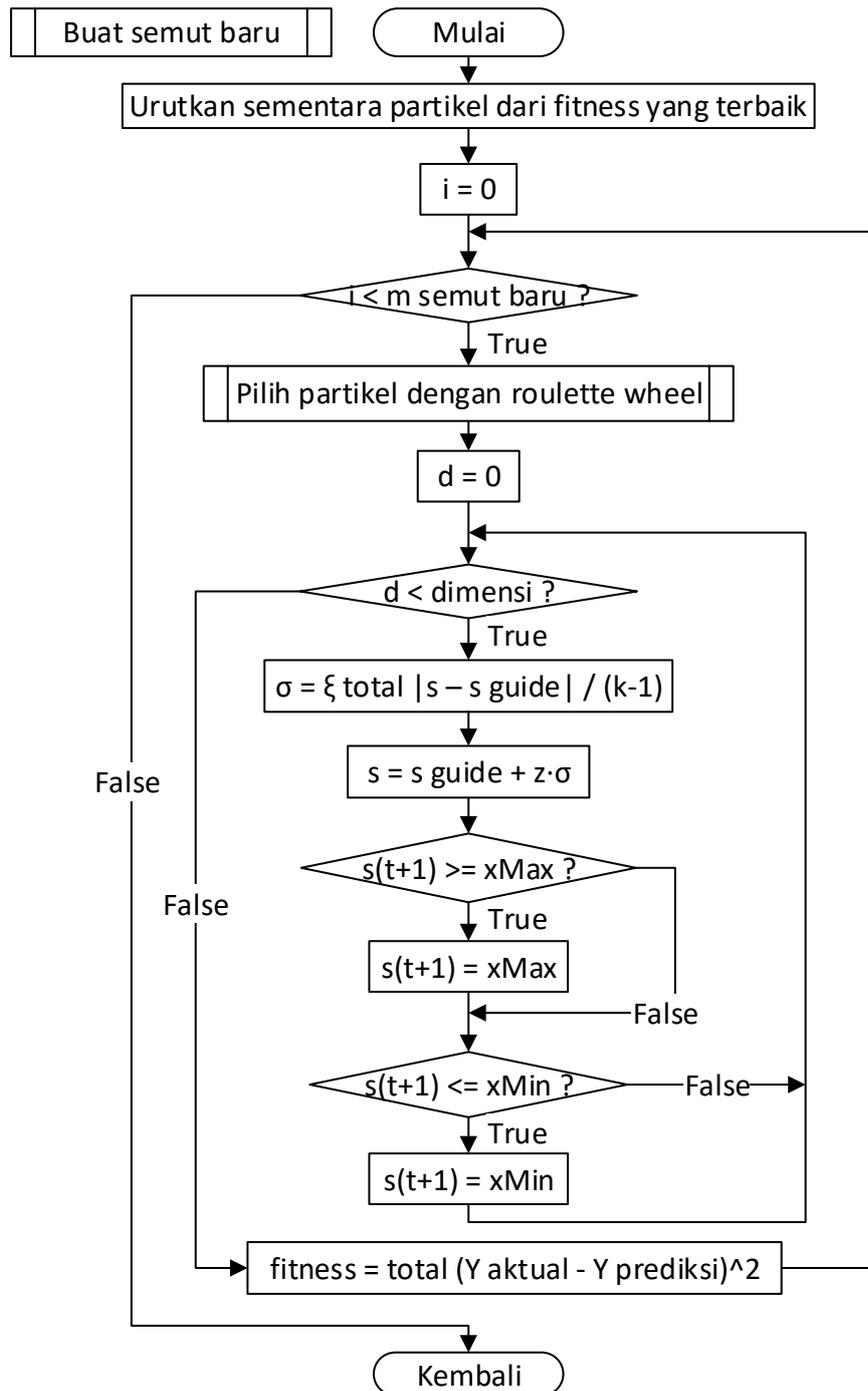


Gambar 4.10 Update gbest tahap PSO

4.1.11 Buat Semut Baru

Buat semut baru merupakan tahap dari ACO_R sendiri. Langkah-langkah membuat semut baru pada ACO_R sesuai dengan flowchart pada Gambar 4.11 yakni:

1. Urutkan tabel partikel secara sementara dimulai dari partikel dengan *fitness* terkecil ke yang terbesar
2. Untuk setiap semut baru pilih partikel dengan cara *roulette wheel* yang ditunjukkan pada Gambar 4.12
3. Hitung sigma masing-masing dimensi semut baru menggunakan Persamaan 2.13
4. Tentukan nilai *random z* yang terdistribusi secara normal (*normally distributed*) kemudian hitung nilai dimensi semut baru dengan Persamaan 2.15
5. Setelah nilai dimensi semut baru dihitung, selanjutnya diperiksa dahulu menggunakan Persamaan 2.15 berupa pembatasan posisi semut baru. Jika kurang dari batas bawah x_{min} dimensi ke-d maka lakukan penggantian nilai dengan x_{min} , sebaliknya jika melebihi x_{max} maka ganti dengan nilai x_{max}
6. Hitung nilai *fitness* semut baru menggunakan Persamaan 2.21

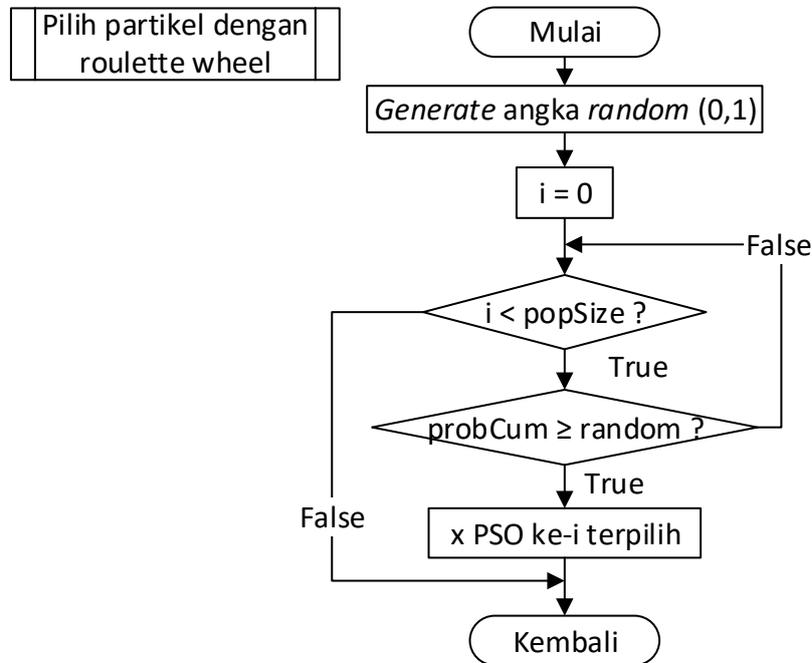


Gambar 4.11 Flowchart buat semut baru

Proses seleksi untuk memilih partikel sebagai s_g menggunakan metode seleksi *roulette wheel* (Fetanat & Khorasaninejad, 2015). Berikut langkah-langkah sesuai dengan flowchart pada Gambar 4.12 yakni:

1. Generate angka *random* antara 0 sampai 1 yang terdistribusi secara seragam (*uniformly distributed*)

- Untuk setiap partikel bandingkan nilai *random* tersebut dengan peluang kumulatif. Jika peluang kumulatif lebih besar atau sama dengan nilai *random* maka pilih partikel ke-*i* yang sejajar dengan peluang kumulatif tersebut dan hentikan pencarian.

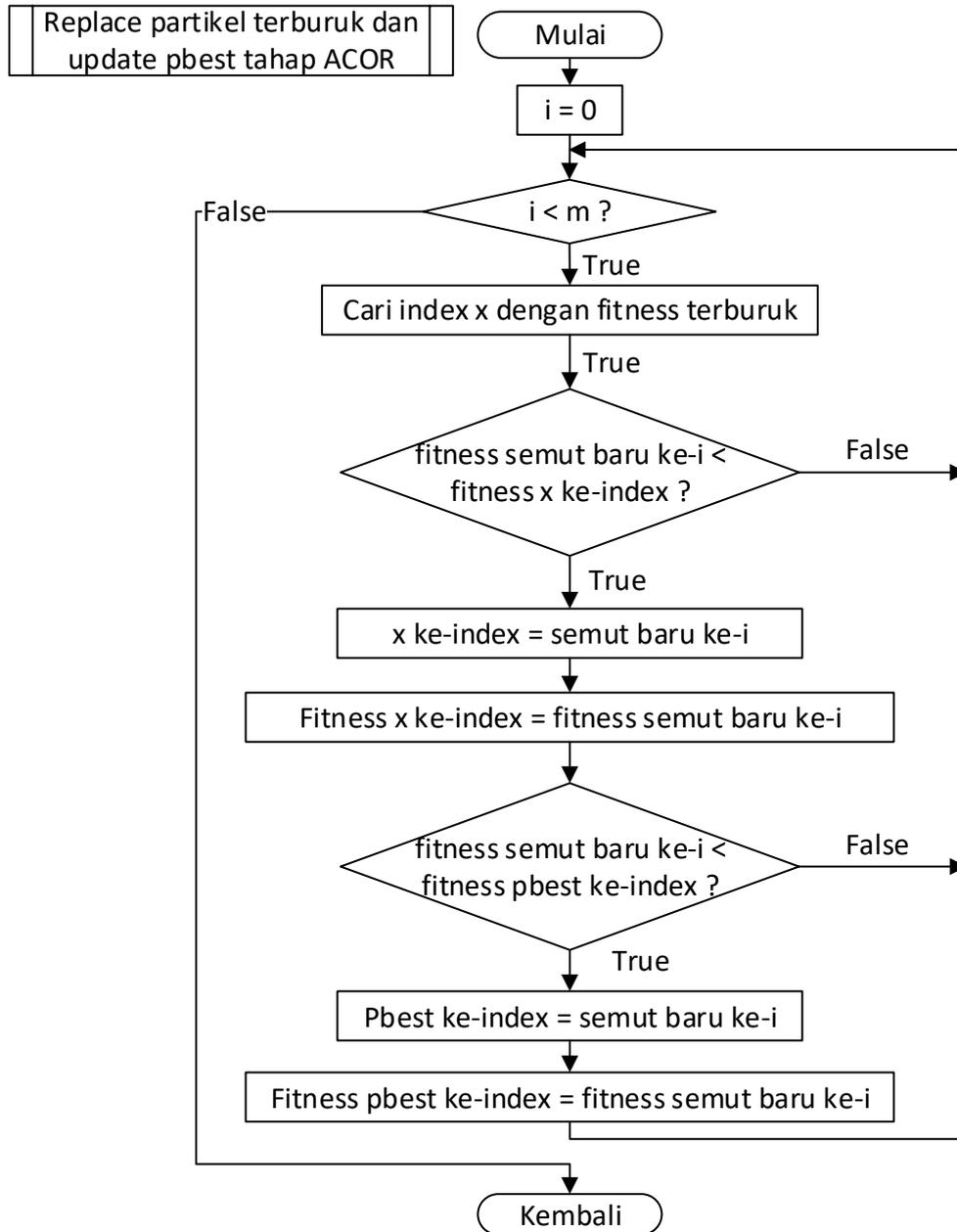


Gambar 4.12 Flowchart pilih partikel dengan *roulette wheel*

4.1.12 *Replace* partikel terburuk dan *update pbest* tahap ACO_R

Setelah PSO meng-*update* pbest dan gbest, ACO_R membuat semut baru kemudian me-*replace* partikel terburuk pada tabel partikel dengan semut baru menggunakan Persamaan 2.17 dan meng-*update* pbest ke-*i* yang sejajar dengan partikel terburuk menggunakan Persamaan 2.18. Jika nilai *fitness* semut baru lebih baik daripada *fitness* partikel terburuk dalam tabel partikel maka lakukan penggantian set solusi hanya pada partikel terburuk, kemudian bandingkan lagi jika *fitness* semut baru lebih baik dari *fitness* pbest ke-*i* yang sejajar dengan partikel terburuk maka lakukan *update* gbest tahap ACO_R hanya pada pbest ke-*i* tersebut.

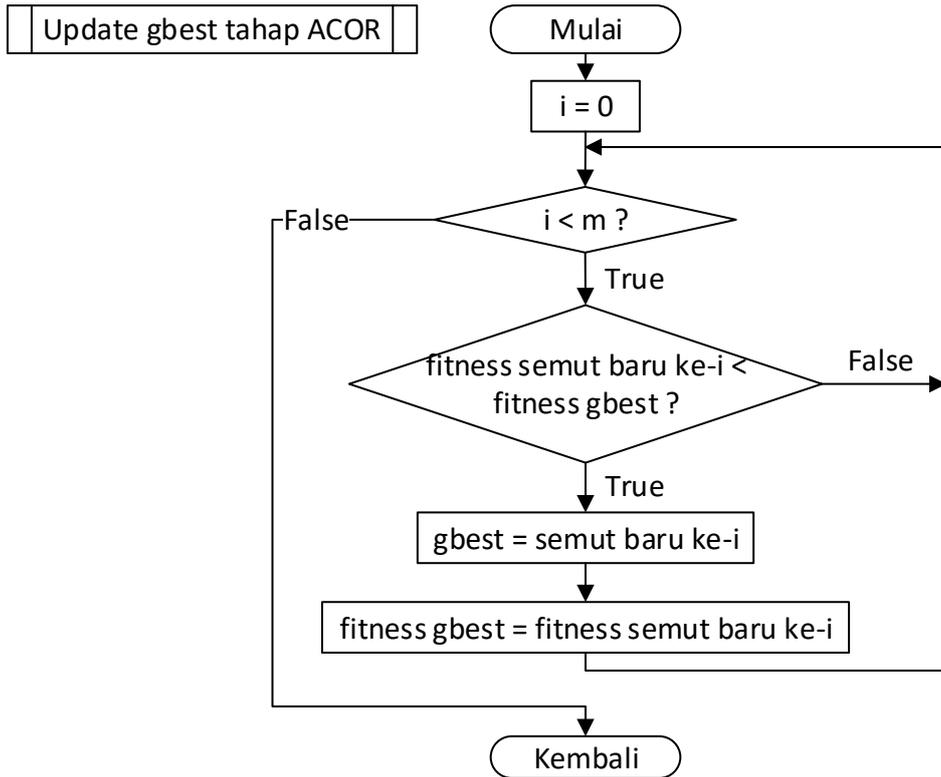
Sebelum proses *replace* partikel terburuk dan *update* pbest tahap ACO_R , lakukan terlebih dahulu proses pencarian index partikel terburuk. Partikel terburuk yang dimaksud adalah partikel yang mempunyai *fitness* terbesar diantara *fitness* partikel lainnya. Proses *replace* partikel terburuk dan *update* pbest tahap ACO_R ditunjukkan pada Gambar 4.13.



Gambar 4.13 Flowchart *replace* partikel terburuk dan *update* gbest tahap ACO_R

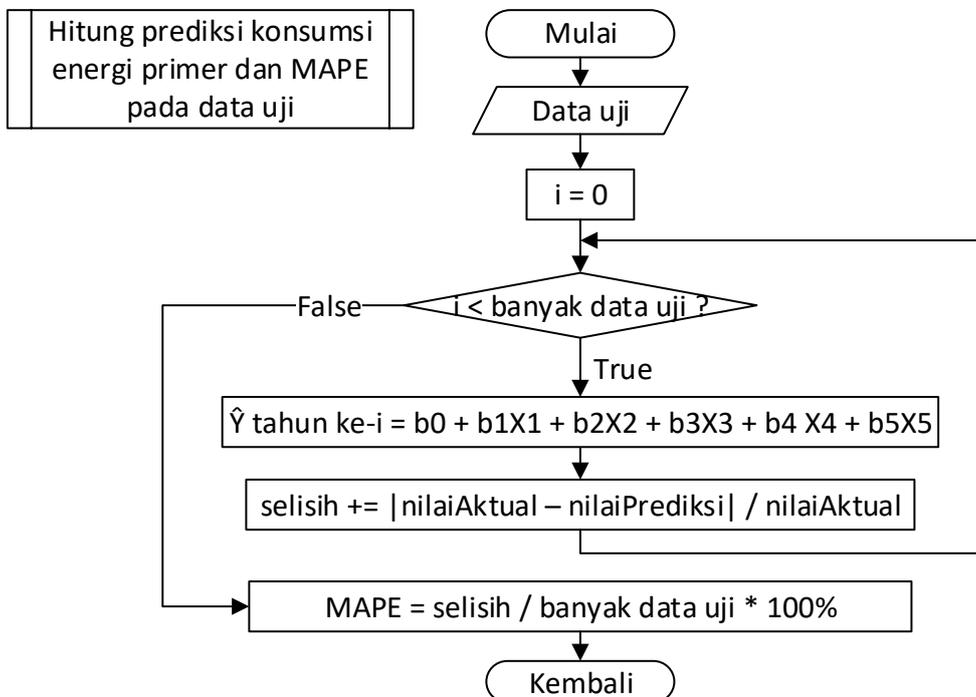
4.1.13 *Update* gbest tahap ACO_R

Proses *update* gbest tahap ACO_R prosesnya hampir sama dengan proses *replace* partikel terburuk dan *update* pbest tahap ACO_R , perbedaannya hanya *fitness* semut baru dibandingkan dengan *fitness* gbest hasil *update* gbest tahap PSO. Jika semut baru mempunyai *fitness* lebih baik daripada *fitness* gbest yang ditemukan pada proses *update* gbest tahap PSO maka lakukan proses *update* berupa penggantian set solusi gbest dengan set solusi semut baru. Proses *update* gbest tahap ACO_R sesuai dengan yang ditunjukkan pada Persamaan 2.19. Untuk lebih jelasnya, proses *update* gbest tahap ACO_R ditunjukkan flowchart pada Gambar 4.14.



Gambar 4.14 Flowchart *update gbest* tahap ACOR

4.1.14 Hitung prediksi konsumsi energi primer dan MAPE pada data uji



Gambar 4.15 Flowchart hitung prediksi konsumsi energi dan MAPE data uji

Proses terakhir adalah menghitung nilai prediksi konsumsi energi primer serta menentukan tingkat *error rate* dengan menggunakan MAPE pada data uji yakni data antara tahun 2007-2016. Proses ini dapat dilihat pada Gambar 4.15.

4.2 Perhitungan Manual

4.2.1 Inisialisasi Parameter PSOACO_R

Tabel 4.1 Parameter PSOACO_R

Parameter	Nilai
w	0,7298
$c1$ dan $c2$	1.496
q	0,2
ξ	0,8
M	1
$popSize$	10
max iterasi	2
min konvergen	10
ruang pencarian intersep (b_0)	[-100, 100]
ruang pencarian koefisien kemiringan (b_1, b_2, b_3, b_4, b_5)	[-10, 10]
$random\ r1$	0.27
$random\ r2$	0.86
$random\ roulette\ wheel$	0.42
$random\ z$	0.87

Parameter PSOACO_R diinisialisasi sesuai dengan nilai parameter optimal yang diuji pada bab 6 agar prediksi menghasilkan nilai *fitness* sekecil mungkin pada data latih dan memiliki efek semakin baik nilai MAPE pada data uji. Parameter-parameter PSOACO_R yang dipakai pada perhitungan manual ini dapat dilihat pada Tabel 4.1. Data latih yang digunakan mulai dari tahun 1967-2006, sedangkan data uji mulai dari tahun 2007 sampai 2016.

4.2.2 Inisialisasi Kecepatan Partikel

Tabel 4.2 Kecepatan awal partikel

v_i	b_0	b_1	b_2	b_3	b_4	b_5
$v_1(0)$	0	0	0	0	0	0
$v_2(0)$	0	0	0	0	0	0
$v_3(0)$	0	0	0	0	0	0
$v_4(0)$	0	0	0	0	0	0
$v_5(0)$	0	0	0	0	0	0
$v_6(0)$	0	0	0	0	0	0
$v_7(0)$	0	0	0	0	0	0
$v_8(0)$	0	0	0	0	0	0
$v_9(0)$	0	0	0	0	0	0
$v_{10}(0)$	0	0	0	0	0	0

Pada tahap inisialisasi atau iterasi = 0, nilai kecepatan awal semua partikel dibuat sama dengan 0 seperti yang ditunjukkan pada Tabel 4.2. Karena bernilai 0 maka hampir semua literatur meniadakan proses inisialisasi kecepatan ini.

Dalam implementasi PSO, terkadang ditemukan bahwa kecepatan partikel bergerak ke nilai yang besar dengan cepat atau biasa disebut dengan *velocity explosion*, terutama untuk posisi partikel yang jauh dari posisi pbest dan individu terbaik (gbest). Akibatnya, partikel tersebut memiliki kecenderungan untuk meninggalkan (keluar) dari ruang batas pencarian atau posisi baru partikel berada di luar range ruang pencarian tersebut.

Oleh karena itu, untuk mengontrol eksplorasi global partikel, perlu adanya pembatasan kecepatan minimum dan maksimum. Teknik pembatasan ini disebut *velocity clamping* untuk mencegah partikel bergerak terlalu jauh melampaui ruang pencariannya. Sebelum melakukan proses *velocity clamping*, terlebih dahulu perlu menentukan batas atas kecepatan atau v_{max} dan batas bawahnya atau $-v_{max}$, berikut perhitungan untuk menentukan batas atas dan bawah *velocity clamping* sesuai dengan Persamaan 2.4:

$$v_{max}^d = 0.6 * (100 - (-100)) / 2$$

$$v_{max}^d = 60.00$$

Jadi $v_i^d \in [-v_{max}, v_{max}]$ atau batas bawah dan atas untuk b_0 yakni [-60, 60]. Nilai tersebut hanya untuk batas bawah dan atas untuk intersep (b_0), sedangkan untuk koefisien kemiringan regresi dapat dilihat pada Tabel 4.3

Tabel 4.3 Batas atas dan bawah *velocity clamping*

v	b₀	b₁	b₂	b₃	b₄	b₅
-v_{max}	-60.00	-6.00	-6.00	-6.00	-6.00	-6.00
v_{max}	60.00	6.00	6.00	6.00	6.00	6.00

4.2.3 Inisialisasi Partikel

Tabel 4.4 Inisialisasi partikel

x_i	b₀	b₁	b₂	b₃	b₄	b₅	fitness
x₁(0)	-63.27	-1.62	1.33	-1.77	3.39	1.87	3175953.0
x₂(0)	-61.16	0.63	4.88	-4.03	-2.42	0.74	5242168.6
x₃(0)	-41.39	2.01	-5.27	3.22	3.16	-4.75	2982611.3
x₄(0)	-62.10	-3.80	-3.27	3.58	2.39	2.01	7099173.0
x₅(0)	-57.70	2.71	-0.58	-1.23	1.54	0.62	975210.9
x₆(0)	-46.07	2.26	0.13	1.58	0.09	-4.25	3707780.5
x₇(0)	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3
x₈(0)	-73.80	0.34	0.89	2.71	-3.36	-1.40	4308026.3
x₉(0)	-75.73	-5.76	-2.61	6.14	0.56	0.99	9952193.9
x₁₀(0)	-54.32	9.39	-3.63	-3.73	6.90	-1.27	7577835.7

Tahap inialisasi partikel PSOACO_R sama persis seperti PSO biasa yakni menggunakan Persamaan 2.2. Berikut merupakan perhitungan nilai partikel awal pada pada x_1 dimensi b_0 :

$$x_1^1 = -100 + 0.18365 * (100 - (-100))$$

$$x = -63.27$$

Untuk menghitung *fitness* pada partikel x_1 perhitungannya sesuai dengan Persamaan 2.21 berupa selisih kuadrat antara nilai aktual dengan nilai prediksi, yakni:

$$fitness = 65036.3 + 67596.7 + 70830.7 + 73313.0 + 75740.4 \dots + 21418.8$$

$$fitness = 3175953.0$$

Hasil lengkap inialisasi partikel untuk semua dimensi permasing-masing partikel dapat dilihat pada Tabel 4.4.

4.2.4 Inialisasi Pbest

Jika masih tahap inialisasi ($t = 0$), maka nilai pbest akan disamakan dengan nilai posisi awal partikel, sedangkan gbest dipilih dari pbest dengan *fitness* terbaik. Hasil inialisasi pbest secara lengkap dapat dilihat pada Tabel 4.5.

Tabel 4.5 Inialisasi pbest

pbest _i	b ₀	b ₁	b ₂	b ₃	b ₄	b ₅	fitness
pbest ₁ (0)	-63.27	-1.62	1.33	-1.77	3.39	1.87	3175953.0
pbest ₂ (0)	-61.16	0.63	4.88	-4.03	-2.42	0.74	5242168.6
pbest ₃ (0)	-41.39	2.01	-5.27	3.22	3.16	-4.75	2982611.3
pbest ₄ (0)	-62.10	-3.80	-3.27	3.58	2.39	2.01	7099173.0
pbest ₅ (0)	-57.70	2.71	-0.58	-1.23	1.54	0.62	975210.9
pbest ₆ (0)	-46.07	2.26	0.13	1.58	0.09	-4.25	3707780.5
pbest ₇ (0)	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3
pbest ₈ (0)	-73.80	0.34	0.89	2.71	-3.36	-1.40	4308026.3
pbest ₉ (0)	-75.73	-5.76	-2.61	6.14	0.56	0.99	9952193.9
pbest ₁₀ (0)	-54.32	9.39	-3.63	-3.73	6.90	-1.27	7577835.7

4.2.5 Inialisasi Gbest

Inialisasi gbest dicari dengan menemukan pbest dengan nilai *fitness* terbaik. Hasil inialisasi gbest ditunjukkan pada Tabel 4.6.

Tabel 4.6 Inialisasi gbest

gbest	b ₀	b ₁	b ₂	b ₃	b ₄	b ₅	fitness
gbest(0)	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3

4.2.6 Hitung ω dan p untuk roulette wheel

Proses menghitung nilai ω dan hitung peluang p_i digunakan untuk proses pemilihan partikel menggunakan roulette wheel, yang mana partikel terpilih akan

digunakan sebagai acuan untuk proses buat semut baru. Berikut perhitungan nilai ω_1 sesuai dengan Persamaan 2.11 dan hitung peluang p_1 sesuai Persamaan 2.12.

$$\omega_1 = \frac{1}{0.2 \cdot 10\sqrt{2\pi}} e^{-\frac{(1-1)^2}{2} \cdot 2^2 \cdot 10^2}$$

$$\omega_1 = \frac{1}{2\sqrt{2\pi}} e^{-\frac{0}{2} \cdot 2^2 \cdot 10^2}$$

$$\omega_1 = 0.1994711402 \cdot 1$$

$$\omega_1 = 0.1994711402$$

$$p_1 = \frac{0.1994711402}{0.2265335416}$$

$$p_1 = 0.880536890$$

Hasil perhitungan secara lengkap ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hitung ω dan p

x_i	ω	p	pCumulative
x_1	0.1994711402	0.880536890	0.880536890
x_2	0.0269954833	0.119167709	0.999704599
x_3	0.0000669151	0.000295387	0.999999987
x_4	0.0000000030	0.000000013	1.000000000
x_5	0.0000000000	0.000000000	1.000000000
x_6	0.0000000000	0.000000000	1.000000000
x_7	0.0000000000	0.000000000	1.000000000
x_8	0.0000000000	0.000000000	1.000000000
x_9	0.0000000000	0.000000000	1.000000000
x_{10}	0.0000000000	0.000000000	1.000000000

4.2.7 Update Kecepatan

Proses hitung *update* kecepatan untuk variabel b_0 pada partikel 1 sesuai dengan Persamaan 2.3.

$$v_1^1(t+1) = 0.729 * 0 + 1.492 * 0.27 * (63,27 - (-63,27)) + 1.492 * 0.86 * (-59.50 - (-63.27))$$

$$v_1^1(t+1) = 4.84$$

Tabel 4.8 Update kecepatan

v_i	b_0	b_1	b_2	b_3	b_4	b_5
$v_1(1)$	4.84	5.61	-7.43	2.17	0.06	0.69
$v_2(1)$	2.13	2.72	-11.98	5.07	7.52	2.14
$v_3(1)$	-23.24	0.95	1.04	-4.23	0.36	9.19
$v_4(1)$	3.34	8.40	-1.53	-4.70	1.35	0.51
$v_5(1)$	-2.31	0.05	-4.98	1.48	2.44	2.30
$v_6(1)$	-17.23	0.63	-5.89	-2.13	4.30	8.55
$v_7(1)$	0.00	0.00	0.00	0.00	0.00	0.00
$v_8(1)$	18.35	3.09	-6.86	-3.58	8.73	4.89
$v_9(1)$	20.83	10.92	-2.37	-7.98	3.70	1.82
$v_{10}(1)$	-6.65	-8.52	-1.06	4.68	-4.44	4.72

Hasil *update* kecepatan pada Tabel 4.8 harus diperbaiki terlebih dengan menggunakan pembatasan kecepatan (*velocity clamping*) agar partikel baru nanti tidak mengalami konvergensi dini maupun terjebak pada local minimum. Pada hasil *update* kecepatan v_1^{b2} diperiksa sesuai Persamaan 2.5 dengan membandingkan dengan tabel batas bawah dan atas *velocity clamping* pada Tabel 4.3. Hasil setelah dilakukan *velocity clamping* ditunjukkan pada Tabel 4.9.

$$v_1^{b2}(t+1) = \begin{cases} v_{max}^{b2}, & -7.43 \geq 6 \\ -v_{max}^{b2}, & -7.43 \leq -6 \\ v_1^{b2}(t+1), & \text{lainnya} \end{cases}$$

$$v_1^{b2}(t+1) = -6$$

Tabel 4.9 Velocity clamping

v_i	b0	b1	b2	b3	b4	b5
$v_1(1)$	4.84	5.61	-6.00	2.17	0.06	0.69
$v_2(1)$	2.13	2.72	-6.00	5.07	6.00	2.14
$v_3(1)$	-23.24	0.95	1.04	-4.23	0.36	6.00
$v_4(1)$	3.34	6.00	-1.53	-4.70	1.35	0.51
$v_5(1)$	-2.31	0.05	-4.98	1.48	2.44	2.30
$v_6(1)$	-17.23	0.63	-5.89	-2.13	4.30	6.00
$v_7(1)$	0.00	0.00	0.00	0.00	0.00	0.00
$v_8(1)$	18.35	3.09	-6.00	-3.58	6.00	4.89
$v_9(1)$	20.83	6.00	-2.37	-6.00	3.70	1.82
$v_{10}(1)$	-6.65	-6.00	-1.06	4.68	-4.44	4.72

4.2.8 Update Posisi Partikel

Setelah kecepatan baru diperoleh maka *update* posisi partikel dapat dihitung. Berikut perhitungan *update* posisi partikel $x_1^1(t+1)$:

$$x_1^1(t+1) = -63.27 + 4.84$$

$$x_1^1(t+1) = -58.43$$

Hasil perhitungan partikel baru iterasi ke-1 dapat dilihat pada Tabel 4.10.

Tabel 4.10 Update posisi partikel

x_i	b0	b1	b2	b3	b4	b5	fitness
$x_1(1)$	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8
$x_2(1)$	-59.03	3.35	-1.12	1.04	3.58	2.88	14361874.9
$x_3(1)$	-64.63	2.96	-4.23	-1.01	3.52	1.25	3774290.8
$x_4(1)$	-58.76	2.20	-4.80	-1.12	3.74	2.52	7376731.8
$x_5(1)$	-60.01	2.76	-5.56	0.25	3.98	2.92	1728975.3
$x_6(1)$	-63.30	2.89	-5.76	-0.55	4.39	1.75	5966816.1
$x_7(1)$	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3
$x_8(1)$	-55.45	3.43	-5.11	-0.87	2.64	3.49	2955576.8
$x_9(1)$	-54.90	0.24	-4.98	0.14	4.26	2.81	8791181.5
$x_{10}(1)$	-60.97	3.39	-4.69	0.95	2.46	3.45	268884.1

4.2.9 Update Pbest Tahap PSO

Untuk *update* pbest, disini kita harus membandingkan antara pbest pada iterasi sebelumnya dengan hasil dari *update* posisi yakni membandingkan Tabel 4.5 dengan Tabel 4.10.

Dari 2 tabel di atas, cek dari urutan baris yang sama, kemudian dibandingkan *fitness*-nya, manakah *fitness* yang lebih baik nilainya akan menjadi pbest terbaru, hasil *update* pbest tahap PSO ditunjukkan pada Tabel 4.11.

Tabel 4.11 Update pbest tahap PSO

pbest_i	b₀	b₁	b₂	b₃	b₄	b₅	fitness
pbest₁(1)	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8
pbest₂(1)	-61.16	0.63	4.88	-4.03	-2.42	0.74	5242168.6
pbest₃(1)	-41.39	2.01	-5.27	3.22	3.16	-4.75	2982611.3
pbest₄(1)	-62.10	-3.80	-3.27	3.58	2.39	2.01	7099173.0
pbest₅(1)	-57.70	2.71	-0.58	-1.23	1.54	0.62	975210.9
pbest₆(1)	-46.07	2.26	0.13	1.58	0.09	-4.25	3707780.5
pbest₇(1)	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3
pbest₈(1)	-55.45	3.43	-5.11	-0.87	2.64	3.49	2955576.8
pbest₉(1)	-54.90	0.24	-4.98	0.14	4.26	2.81	8791181.5
pbest₁₀(1)	-60.97	3.39	-4.69	0.95	2.46	3.45	268884.1

4.2.10 Update Gbest Tahap PSO

Untuk *update* gbest PSO diperoleh dari pbest tahap PSO yang memiliki *fitness* terbaik. Jika melihat tabel pbest pada Tabel 4.11 maka dapat diketahui bahwa gbest tahap PSO adalah pbest pertama. Hasil *update* gbest tahap PSO ditunjukkan pada Tabel 4.12.

Tabel 4.12 Update gbest tahap PSO

gbest	b₀	b₁	b₂	b₃	b₄	b₅	fitness
gbest(1)	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8

4.2.11 Buat Semut Baru

Proses pertama buat semut baru berupa pengurutan tabel partikel secara sementara agar sesuai dengan perankingan nilai ω dan peluang p , dimana partikel dengan *fitness* terbaik memiliki peluang terpilih yang lebih besar dibanding yang lain. Hasil pengurutan tabel partikel ditunjukkan pada Tabel 4.13.

Pilih Partikel dengan Roulette Wheel

Misal random untuk roulette wheel adalah = 0.42 maka partikel terpilih adalah partikel yang pertama. Nilai partikel yang diperoleh menggunakan roulette wheel ditunjukkan pada Tabel 4.14.

Tabel 4.13 Urutkan sementara tabel partikel mulai dari fitness terbaik

x_1	b_0	b_1	b_2	b_3	b_4	b_5	fitness
$x_1(1)$	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8
$x_2(1)$	-60.97	3.39	-4.69	0.95	2.46	3.45	268884.1
$x_3(1)$	-59.50	2.75	-4.46	-0.08	3.44	2.41	963196.3
$x_4(1)$	-60.01	2.76	-5.56	0.25	3.98	2.92	1728975.3
$x_5(1)$	-55.45	3.43	-5.11	-0.87	2.64	3.49	2955576.8
$x_6(1)$	-64.63	2.96	-4.23	-1.01	3.52	1.25	3774290.8
$x_7(1)$	-63.30	2.89	-5.76	-0.55	4.39	1.75	5966816.1
$x_8(1)$	-58.76	2.20	-4.80	-1.12	3.74	2.52	7376731.8
$x_9(1)$	-54.90	0.24	-4.98	0.14	4.26	2.81	8791181.5
$x_{10}(1)$	-41.16	3.91	0.88	5.03	4.00	3.33	59386.97

Tabel 4.14 Pilih partikel dengan roulette wheel

x_i	b_0	b_1	b_2	b_3	b_4	b_5
$x_1(1)$	-58.43	3.99	-4.67	0.40	3.45	2.56

Hitung Sigma Partikel Terpilih

Setelah random solusi terpilih kemudian hitung sigma. Perhitungan nilai sigma pada dimensi b_0 pada semut baru yakni:

$$\sigma_l^1 = 0,8 * 2.63$$

$$\sigma_l^i = 2.10$$

Hasil lengkap perhitungan sigma untuk setiap dimensi dapat ditunjukkan pada Tabel 4.15.

Tabel 4.15 Hitung sigma

sigma _i	b_0	b_1	b_2	b_3	b_4	b_5
sigma ₁ (1)	2.10	1.06	0.63	0.64	0.40	0.45

Tabel 4.16 Nilai z

z	b_0	b_1	b_2	b_3	b_4	b_5
z	0.87	0.87	0.87	0.87	0.87	0.87

Hitung Semut Baru

Setelah menghitung sigma partikel terpilih maka selanjutnya menghitung semut baru:

$$s_1^1 = 2.10 + 0.87 * -58.43$$

$$s_1^1 = -56.6$$

Pada perhitungan tersebut nilai Z seperti pada Tabel 4.16 dibuat sama agar memudahkan menyamakan hasil dengan keluaran perangkat lunak. Hasil perhitungan nilai semut baru setiap dimensinya dapat dilihat pada Tabel 4.17.

Tabel 4.17 Semut Baru

s_m	b_0	b_1	b_2	b_3	b_4	b_5	fitness
$s_1(1)$	-56.60	4.91	-4.12	0.96	3.81	2.95	6109817.8

4.2.12 Replace partikel terburuk dan update pbest tahap ACO_R

Jika semut baru yang diperoleh mempunyai *fitness* yang lebih baik daripada partikel terburuk pada proses sebelumnya yakni *update* pbest tahap PSO maka ganti partikel tersebut dengan semut baru, lalu lakukan *update* pbest. Seperti yang ditunjukkan pada Tabel 4.18, jika *fitness* partikel terburuk lebih buruk dari *fitness* semut baru maka perlu dilakukan proses *replace* antara partikel dengan *fitness* terburuk dengan semut baru tersebut. Jika semut baru tidak lebih baik *fitness*-nya dari partikel *fitness* terburuk maka tidak perlu melakukan proses *replace* partikel terburuk maupun *update* pbest tahap ACO_R. Hal ini sesuai dengan Persamaan 2.17.

Tabel 4.18 Replace partikel terburuk dengan semut baru

	b_0	b_1	b_2	b_3	b_4	b_5	fitness
$x_2(1)$	-59.03	3.35	-1.12	1.04	3.58	2.88	14361874.9
vs.							
$s_1(1)$	-56.60	4.91	-4.12	0.96	3.81	2.95	6109817.8
=							
$x_2(1)$	-56.60	4.91	-4.12	0.96	3.81	2.95	6109817.8

Saat kondisi *fitness* semut baru lebih baik daripada *fitness* partikel terburuk maka lakukan juga proses *update* pbest tahap ACO_R sesuai dengan Persamaan 2.18. Proses *update* pbest tahap ACO_R hanya meng*update* pbest yang sejajar dengan partikel terburuk yang ditemukan. Untuk lebih jelasnya dapat dilihat pada Tabel 4.19.

Tabel 4.19 Update pbest tahap ACO_R

	b_0	b_1	b_2	b_3	b_4	b_5	fitness
$pbest_2(1)$	-61.16	0.63	4.88	-4.03	-2.42	0.74	5242168.6
vs.							
$s_1(1)$	-56.60	4.91	-4.12	0.96	3.81	2.95	6109817.8
=							
$pbest_2(1)$	-61.16	0.63	4.88	-4.03	-2.42	0.74	5242168.6

4.2.13 Update gbest tahap ACO_R

Hal yang sama juga dilakukan dengan membandingkan antara nilai *fitness* gbest tahap PSO dengan *fitness* semut baru seperti yang ditunjukkan pada Tabel 4.20. Sesuai dengan Persamaan 2.19 yakni jika *fitness* semut baru lebih baik maka ganti gbest tahap PSO pada iterasi tersebut, jika tidak maka gbest tetap yakni menggunakan gbest tahap PSO.

Tabel 4.20 Update gbest tahap ACO_R

	b₀	b₁	b₂	b₃	b₄	b₅	fitness
gbest(1)	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8
vs.							
s₁(1)	-56.60	4.91	-4.12	0.96	3.81	2.95	6109817.8
=							
gbest(1)	-58.43	3.99	-4.67	0.40	3.45	2.56	241325.8

4.2.14 Hitung prediksi konsumsi energi primer dan hitung MAPE pada data uji

Proses terakhir adalah menghitung nilai prediksi konsumsi energi primer serta menentukan tingkat *error rate* dengan menggunakan MAPE pada data uji yakni data antara tahun 2007-2016. Hasil pemodelan linear menggunakan PSOACO_R pada iterasi 1 yakni

$$Y = -58.43 + 3.99X_1 - 4.67X_2 + 0.40X_3 + 3.45X_4 + 2.56X_5$$

Tabel 4.21 Hasil prediksi

Tahun	nilai aktual	nilai prediksi	selisih
2007	132.87	187.70	54.83
2008	131.3	199.86	68.56
2009	136.01	213.65	77.64
2010	149.31	208.94	59.63
2011	162.83	146.67	16.16
2012	170.54	216.48	45.94
2013	174.24	224.78	50.54
2014	162.9	213.62	50.72
2015	164.83	204.20	39.37
2016	175.04	264.70	89.66

Setelah pemodelan regresi linear terbentuk, maka dilakukan proses perhitungan MAPE pada data uji sesuai dengan Persamaan 2.22 pada iterasi pertama sebesar:

$$MAPE = \frac{0.41 + 0.52 + 0.57 + 0.40 + 0.10 + 0.27 + 0.29 + 0.31 + 0.24 + 0.51}{10} * 100\%$$

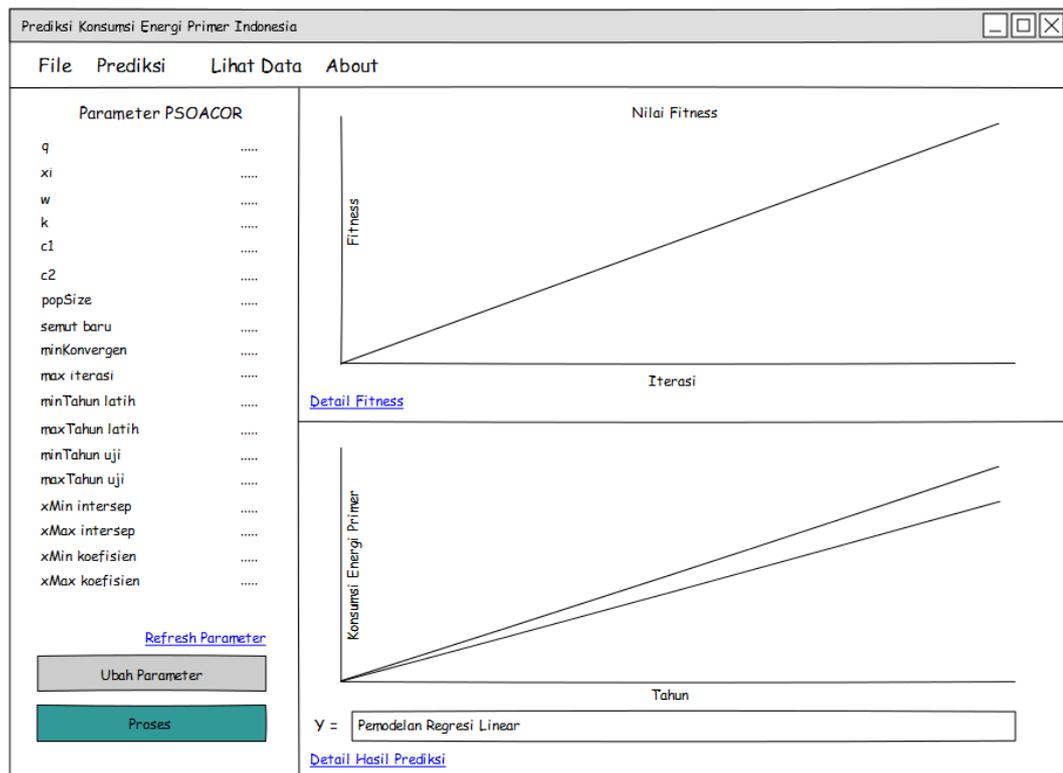
$$MAPE = 36.26\%$$

4.3 Perancangan Antarmuka

Pada perancangan antarmuka pengguna dari perangkat lunak konsumsi energi primer Indonesia bisa dilihat pada Gambar 4.16. Pada pembuatan perangkat lunak ini nantinya akan dibuat 5 halaman antarmuka utama yang terdiri dari antarmuka prediksi, antarmuka ubah parameter, antarmuka detail *fitness*, antarmuka detail prediksi, dan antarmuka lihat semua data.

4.3.1 Perancangan Antarmuka Prediksi

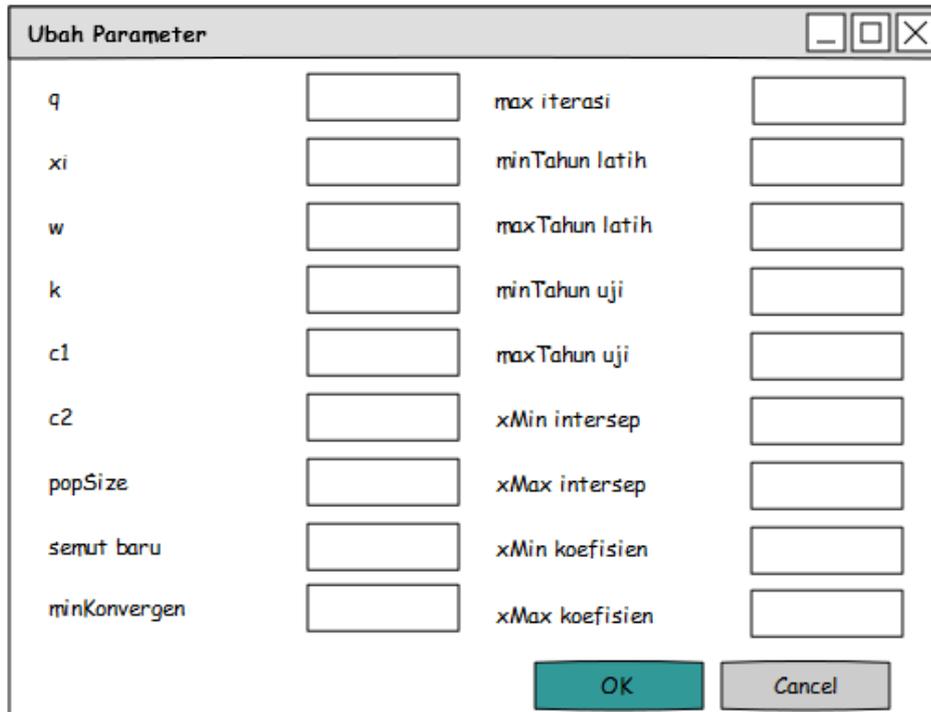
Perancangan antarmuka prediksi adalah halaman yang berisi hasil prediksi beserta nilainya dalam bentuk grafik grafis. Di dalamnya terdapat 3 bagian utama yakni menampilkan parameter PSOACO_R yang digunakan, hasil *fitness* berupa grafik garis, dan hasil prediksi dalam bentuk grafik garis juga. Rancangan antarmuka prediksi konsumsi energi primer Indonesia dapat dilihat pada Gambar 4.16.



Gambar 4.16 Perancangan antarmuka prediksi dengan PSOACO_R

4.3.2 Perancangan Antarmuka Ubah Parameter

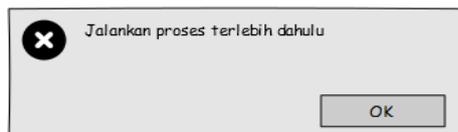
Pada perancangan antarmuka pengujian berisi antarmuka yang akan menampilkan parameter-parameter apa saja yang bisa diubah yakni parameter q , xi , w , k *velocity clamping*, $c1$, $c2$, $popSize$, m semut baru, min konvergen, max iterasi, tahun latih tahun uji, ruang pencarian intersep dan koefisien kemiringan regresi. Antarmuka ubah parameter hanya bisa diakses setelah pengguna menekan button Ubah Parameter (Gambar 4.16). Jika pengguna sudah mengganti nilai parameter seperti yang diinginkan, tekan tombol OK untuk menyimpan perubahan. Parameter yang sudah diganti akan ditampilkan pada antarmuka prediksi setelah pengguna menekan *hyperlink* Refresh Parameter (Gambar 4.16). Perancangan antarmuka ubah parameter dapat dilihat pada Gambar 4.17.



Gambar 4.17 Perancangan antarmuka ubah parameter

4.3.3 Perancangan Antarmuka Detail Fitness

Perancangan antarmuka detail *fitness* pada Gambar 4.19 berisi nilai *fitness* yakni selisih absolut antara nilai aktual dan nilai prediksi yang ditampilkan periterasi mulai dari iterasi ke-0 (inisialisasi) sampai iterasi terakhir dalam bentuk tabel. Antarmuka dapat dibuka jika pengguna sudah menekan tombol proses, menunggu sampai proses prediksi selesai, dan menekan hyperlink detail *fitness* seperti pada Gambar 4.16, jika tidak maka akan tampil antarmuka peringatan seperti pada Gambar 4.18.



Gambar 4.18 Perancangan pesan kesalahan detail fitness

Detail Fitness	
Iterasi	Fitness
0	..
1	..
2	..
3	..
4	..
5	..
6	..

Gambar 4.19 Perancangan antarmuka detail fitness

4.3.4 Perancangan Antarmuka Detail Prediksi

Gambar 4.20 menunjukkan perancangan antarmuka detail prediksi yang didalamnya menampilkan nilai aktual beserta nilai hasil prediksi data uji pertahunnya. Untuk mengakses antarmuka ini pengguna harus menyelesaikan proses prediksi dan menekan hyperlink Detail Prediksi Gambar 4.16 atau jika tidak maka akan tampil antarmuka peringatan seperti Gambar 4.18.

Detail Prediksi			
Tahun	Nilai Aktual	Nilai Prediksi	Selisih
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016

MAPE (%) =
Waktu Komputasi (ms) =

Gambar 4.20 Perancangan antarmuka detail hasil prediksi

4.3.5 Perancangan Antarmuka Lihat Data

Semua data yang dipakai pada penelitian ini baik data latih dan data uji akan ditampilkan pada antarmuka lihat semua data (Gambar 4.21). Data akan ditampilkan pertahun dari tahun 1967-2016 dengan data berupa nilai aktual konsumsi energi primer, GNI, GDP, populasi, impor, dan ekspor Indonesia dalam bentuk tabel.

Setelah proses prediksi konsumsi energi Indonesia, maka akan didapatkan nilai berupa pemodelan regresi linear beserta MAPE-nya. Nilai pemodelan tersebut saling berbeda satu sama lain karena metode yang digunakan untuk mencari intersep dan koefisien kemiringan regresi bersifat *stochastic*. Agar mengetahui pemodelan mana yang mempunyai MAPE terkecil, maka data pemodelan regresi linear disimpan dalam database. Untuk menampilkan data

4.4 Perancangan Pengujian

Pada metode *hybrid* PSO dan ACO_R (PSOACO_R), untuk menentukan yang optimal perlu dilakukan percobaan pada parameter tertentu. Percobaan-percobaan dilakukan berguna untuk menemukan konstanta dan koefisien regresi yang optimal untuk prediksi konsumsi energi primer Indonesia, percobaan-percobaan yang akan dilakukan yakni:

1. Pengujian dan analisis nilai c_1 dan c_2 terhadap *fitness*
2. Pengujian dan analisis nilai w terhadap *fitness*
3. Pengujian dan analisis nilai k terhadap *fitness*
4. Pengujian dan analisis nilai q terhadap *fitness*
5. Pengujian dan analisis nilai ξ terhadap *fitness*
6. Pengujian dan analisis jumlah semut baru terhadap *fitness*
7. Pengujian dan analisis kriteria berhenti terhadap *fitness*
8. Pengujian dan analisis jumlah popsize terhadap *fitness*
9. Pengujian dan analisis pemodelan regresi linear terhadap MAPE
10. Perbandingan antara regresi-PSOACO_R dengan regresi-PSO dan regresi-ACO_R

4.4.1 Perancangan Pengujian Nilai c_1 dan c_2 Terhadap Fitness

Nilai c_1 dan c_2 bersama dengan nilai w , merupakan parameter yang menentukan keseimbangan antara *global exploration* dan *local exploitation* serta mengatur tingkat konvergensi pada PSO. Pada pengujian ini dilakukan percobaan dengan 8 nilai c_1 dan c_2 yang berbeda, dimana masing-masing nilai c_1 dan c_2 bernilai sama untuk masing-masing percobaan. Untuk menganalisis hasil prediksi maka hasil pengujian masing-masing nilai c_1 dan c_2 dihitung rata-rata *fitness*-nya. Tujuan dari pengujian nilai c_1 dan c_2 adalah untuk mengetahui bagaimana pengaruh nilai c_1 dan c_2 terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian nilai c_1 dan c_2 yang ditampilkan pada Tabel 4.22.

Tabel 4.22 Perancangan pengujian nilai c_1 dan c_2

c_1, c_2	Nilai fitness percobaan ke-										Avg. fitness
	1	2	3	4	5	6	7	8	9	10	
1.492, 1.492											
1.5, 1.5											
1.6, 1.6											
1.7, 1.7											
1.8, 1.8											
1.9, 1.9											
2.0, 2.0											
2.05 2.05											

4.4.2 Perancangan Pengujian Nilai w Terhadap Fitness

Nilai w berfungsi untuk mengatur tingkat konvergensi pada tahap PSO. Semakin mendekati 0 maka tingkat konvergensi tahap PSO semakin tinggi yang mengakibatkan terjadinya konvergensi dini. Pada pengujian ini dilakukan percobaan dengan 6 nilai w yang berbeda. Untuk menganalisis hasil prediksi maka hasil pengujian masing-masing nilai w dihitung rata-rata *fitness*-nya. Tujuan

dari pengujian nilai w adalah untuk mengetahui bagaimana pengaruh w terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian nilai w yang ditampilkan pada Tabel 4.23.

Tabel 4.23 Perancangan pengujian nilai w

w	Nilai <i>fitness</i> percobaan ke-										Avg. <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0.5											
0.6											
0.7											
0.729											
0.8											
0.9											

4.4.3 Perancangan Pengujian Nilai k velocity clamping Terhadap *Fitness*

Nilai k pada proses *velocity clamping* berguna untuk membatasi kecepatan partikel agar tidak mudah keluar dari ruang pencarian serta mencegah terjadinya *velocity explosion* pada PSO. Pada pengujian ini dilakukan percobaan dengan 10 nilai k yang berbeda. Untuk menganalisis hasilnya maka hasil pengujian masing-masing nilai k dihitung rata-rata *fitness*-nya. Tujuan dari pengujian nilai k adalah untuk mengetahui bagaimana pengaruh nilai k terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian nilai k yang ditampilkan pada Tabel 4.24.

Tabel 4.24 Perancangan pengujian nilai k velocity clamping

k	Nilai <i>fitness</i> percobaan ke-										Avg. <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
0.1											
0.2											
0.3											
0.4											
0.5											
0.6											
0.7											
0.8											
0.9											
1											

4.4.4 Perancangan Pengujian Nilai q Terhadap *Fitness*

Nilai q berfungsi untuk mengontrol tingkat diversifikasi atau *global exploration* pada ACO_R. Pada pengujian ini dilakukan percobaan dengan 10 nilai q yang berbeda. Untuk menganalisis hasil prediksi maka hasil pengujian nilai q dihitung rata-rata *fitness*-nya. Tujuan dari pengujian nilai q adalah untuk mengetahui bagaimana pengaruh nilai q terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian nilai q yang ditampilkan pada Tabel 4.25.

Tabel 4.25 Perancangan pengujian nilai q

q	Nilai fitness percobaan ke-										Avg. Fitness
	1	2	3	4	5	6	7	8	9	10	
0.001											
0.002											
0.003											
0.004											
0.005											
0.006											
0.007											
0.008											
0.009											
0.01											

4.4.5 Perancangan Pengujian Nilai ξ Terhadap Fitness

Nilai ξ merupakan parameter yang mengontrol tingkat konvergensi pada ACO_R . Pada pengujian ini dilakukan percobaan dengan 10 nilai ξ yang berbeda. Untuk menganalisis hasilnya maka hasil pengujian masing-masing nilai ξ dihitung rata-rata *fitness*-nya. Tujuan dari pengujian nilai ξ adalah untuk mengetahui bagaimana pengaruh nilai ξ terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian nilai ξ yang ditampilkan pada Tabel 4.26.

Tabel 4.26 Perancangan pengujian nilai ξ

ξ	Nilai fitness percobaan ke-										Avg. Fitness
	1	2	3	4	5	6	7	8	9	10	
0.1											
0.2											
0.3											
0.4											
0.5											
0.6											
0.7											
0.8											
0.9											
1											

4.4.6 Perancangan Pengujian Jumlah Semut Baru Terhadap Fitness

Pada pengujian ini dilakukan percobaan dengan 9 jumlah semut yang berbeda. Pengujian jumlah semut baru dimulai dari 2 sampai 50 semut. Untuk menganalisisnya maka hasil pengujian masing-masing jumlah semut dihitung rata-rata *fitness*-nya. Tujuan dari pengujian jumlah semut baru adalah untuk mengetahui bagaimana pengaruh jumlah semut baru terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian jumlah semut baru yang ditampilkan pada Tabel 4.27.

Tabel 4.27 Perancangan pengujian jumlah semut

m	Nilai fitness percobaan ke-										Avg. fitness
	1	2	3	4	5	6	7	8	9	10	
2											
4											
6											
8											
10											
20											
30											
40											
50											

4.4.7 Perancangan Pengujian Jumlah PopSize Terhadap fitness

Pada pengujian ini dilakukan percobaan dengan 10 jumlah popSize yang berbeda, dimana popsize partikel dengan range 20 sampai 200 dengan interval sebanyak 20. Untuk menganalisisnya maka hasil pengujian masing-masing jumlah popSize dihitung rata-rata *fitness*-nya. Tujuan dari pengujian jumlah popSize adalah untuk mengetahui bagaimana pengaruh jumlah popSize terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian jumlah popSize yang ditampilkan pada Tabel 4.28.

Tabel 4.28 Perancangan pengujian jumlah popsize

PopSize	Nilai fitness percobaan ke-										Avg. Fitness
	1	2	3	4	5	6	7	8	9	10	
20											
40											
60											
80											
100											
120											
140											
160											
180											
200											

4.4.8 Perancangan Pengujian Kriteria Berhenti Terhadap Fitness

Pada pengujian ini dilakukan percobaan dengan 10 kriteria berhenti yang berbeda, dimana pada penelitian ini digunakan max iterasi dan minimum konvergen tergantung mana yang tercapai terlebih dahulu. Untuk menganalisis hasil prediksi maka hasil pengujian masing-masing kriteria berhenti dihitung rata-rata *fitness*-nya. Tujuan dari pengujian kriteria berhenti adalah untuk mengetahui bagaimana pengaruh kriteria berhenti terhadap nilai *fitness* pada data latih. Berikut ini adalah rancangan pengujian kriteria berhenti yang ditampilkan pada Tabel 4.29.

Tabel 4.29 Perancangan pengujian kriteria berhenti

iterasi, konvergen	Nilai fitness percobaan ke-										Avg. Fitness
	1	2	3	4	5	6	7	8	9	10	
500, 50											
500, 100											
1000, 50											
1000, 100											
1500, 50											
1500, 100											
2000, 50											
2000, 100											
2500, 50											
2500, 100											

4.4.9 Perancangan Pengujian Pemodelan Regresi Linear Terhadap MAPE

Pengujian pemodelan regresi linear dilakukan setelah parameter-parameter terbaik ditemukan, parameter yang terpilih akan dijalankan untuk prediksi sebanyak 10 kali bertujuan untuk mengetahui apakah perangkat lunak telah valid dengan menganalisis berdasarkan hasil prediksinya. Jika 10 kali percobaan menghasilkan MAPE yang berbeda jauh atau fluktuatif maka perangkat lunak dinyatakan tidak valid dan perlu dilakukan perbaikan dalam proses prediksi, namun jika sebaliknya maka perangkat lunak dinyatakan telah valid dan layak digunakan.

Setelah hasil keluaran yang dihasilkan oleh perangkat lunak tidak fluktuatif, maka tentukan 10 pemodelan regresi linear dengan MAPE data uji terkecil. Dari pemodelan regresi linear tersebut lakukan prediksi konsumsi energi primer pada data uji. Berikut ini adalah rancangan pengujian yang ditampilkan pada Tabel 4.30 beserta perbandingan antara nilai aktual dan nilai prediksi pada 10 pemodelan regresi linear terbaik (mempunyai *fitness* terkecil) yang akan dijabarkan pada Tabel 4.31.

Tabel 4.30 Rancangan pengujian pemodelan regresi-PSOACO_R terhadap MAPE

	Nilai prediksi percobaan ke-									
	1	2	3	4	5	6	7	8	9	10
b0										
b1										
b2										
b3										
b4										
b5										
Fitness										
Kondisi Berhenti										
MAPE data uji										

Tabel 4.31 Perbandingan nilai aktual dan prediksi

Tahun	Nilai aktual	Nilai prediksi percobaan ke-									
		1	2	3	4	5	6	7	8	9	10
2007	132.87										
2008	131.30										
2009	136.01										
2010	149.31										
2011	162.83										
2012	170.54										
2013	174.24										
2014	162.90										
2015	164.83										
2016	175.04										
	MAPE										

4.4.10 Perancangan Perbandingan Hasil antara Regresi-PSOACOR dengan Regresi-PSO, dan Regresi-ACOR

Perbandingan dilakukan untuk mengetahui apakah model regresi lebih baik dengan optimasi menggunakan hybrid ACOR dan PSO atau tidak. Untuk menganalisis hasilnya maka dilakukan pengujian masing-masing regresi dengan metode berbeda kemudian dihitung rata-rata *fitness*-nya. Berikut ini adalah rancangan perbandingan hasil MAPE dan waktu komputasi antara regresi-PSOACOR dengan regresi-PSO dan regresi-ACOR yang ditampilkan pada Tabel 4.32 dan Tabel 4.33.

Tabel 4.32 Rancangan pengujian MAPE antara ACOR, PSO dan PSOACOR

Metode	Nilai MAPE percobaan ke-										Avg. MAPE
	1	2	3	4	5	6	7	8	9	10	
Regresi-PSOACOR											
Regresi-PSO											
Regresi-ACOR											

Tabel 4.33 Rancangan pengujian waktu komputasi antara ACOR, PSO, PSOACOR

Metode	Waktu komputasi percobaan ke-										Avg. (ms)
	1	2	3	4	5	6	7	8	9	10	
Regresi-PSOACOR											
Regresi-PSO											
Regresi-ACOR											