

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Penelitian Terdahulu

Pada banyak literatur, penelitian pada prediksi energi primer telah banyak dilakukan. Prediksi menggunakan *Autoregressive Integrated Moving Average* (ARIMA), *Artificial Neural Network* (ANN), *Grey Model*, *Genetic Algorithm* (GA), logika fuzzy, *Support Vector Regression* (SVR), *Ant Colony Optimization* (ACO), *Particle Swarm Optimization* (PSO), dan *Long-range Energy Alternatives Planning* (LEAP) merupakan beberapa model yang paling banyak digunakan dalam penelitian prediksi energi primer (Suganthi & Samuel, 2012).

Yuan, et al. (2017) memprediksi kebutuhan energi primer per provinsi dan nasional di China menggunakan pendekatan *hierarchical* Bayesian. Data yang digunakan berasal dari 30 provinsi selama kurun waktu 1995-2012 berupa data *Gross Domestic Product* (GDP), populasi, tingkat urbanisasi, konsumsi energi, dan persentase industri terhadap GDP. Hasil penelitian menunjukkan bahwa faktor-faktor tersebut dapat digunakan untuk memprediksi kebutuhan energi per provinsi dan nasional di China. Hasil terbaik dicapai menggunakan pemodelan dengan menggunakan dua faktor berupa tingkat industrialisasi dan tingkat urbanisasi, RMSE terbaik yang didapatkan sebesar 0.024.

Azadeh, et al. (2013) melakukan prediksi konsumsi energi terbarukan di Iran menggunakan pemodelan regresi linear – Multi Layer Perceptron (MLP). Data yang digunakan berupa data bulanan selama 11 tahun (1996-2006) berupa data GDP, karbon dioksida, karbon monoksida, nitrogen oksida, harga gas, dan harga minyak di Iran. MAPE yang didapat menggunakan metode tersebut sebesar 99.9% dan hasilnya lebih baik daripada pemodelan regresi biasa tanpa MLP dan pemodelan regresi – fuzzy.

Barak & Sadegh (2016) menggunakan data GDP, populasi, impor, dan ekspor dari tahun 1967 sampai 2012 untuk memprediksi konsumsi energi primer di Iran menggunakan metode hybrid ARIMA-ANFIS. Hasil pada data uji menunjukkan nilai MSE terkecil sebesar 0.000263.

Deka, et al. (2016) membuat pemodelan untuk memprediksi kebutuhan energi primer Amerika Serikat menggunakan lima teknik yang berbeda yakni dua *artificial neural network* (ANN), dua pemodelan regresi, dan satu *autoregressive integrated moving average* (ARIMA). Data yang digunakan mulai dari tahun 1950-2013 berupa data konsumsi energi primer, GDP, *gross national product* atau *gross national income* (GNI), dan pendapatan per kapita. Hasil menggunakan model ANN pertama merupakan yang terbaik dengan RMSE pada data uji sebesar 1.99.

Yu, et al. (2012) mencoba pendekatan PSO-GA untuk memprediksi kebutuhan energi primer di China menggunakan tiga pemodelan regresi berbeda yakni regresi linear, eksponensial, dan kuadrat. Faktor-faktor yang digunakan untuk melakukan prediksi terdiri dari lima hal, GDP, populasi, persentase industri

terhadap GDP, persentase tingkat urbanisasi terhadap total populasi, dan presentase penggunaan batubara dari total konsumsi energi. Metode yang digunakan menunjukkan hasil PSO-GA lebih baik daripada pemodelan regresi menggunakan satu metode yakni GA, PSO, atau ACO dengan hasil rata-rata MAPE sebesar 0.54% pada data uji.

Uzlu, et al. (2014) juga menggunakan data GDP, populasi, impor, dan ekspor untuk memprediksi konsumsi energi primer di Turki menggunakan metode *feed forward backpropagation-teaching learning based optimization*. Hasil pada data uji menunjukkan nilai RMSE sebesar 2.192.

Kiran, et al. (2012) mengembangkan metode pemodelan regresi linear dan kuadratik menggunakan *hybrid* ACO (bukan kontinyu) dan PSO untuk memprediksi kebutuhan energi primer di Turki. Fungsi *fitness* yang dipakai berupa selisih kuadrat antara nilai aktual dan prediksi pada data latih. Data yang dipakai pada penelitian ini berupa data GDP, populasi, impor, dan ekspor negara Iran. Hasil pengujian menunjukkan metode yang digunakan memiliki nilai relative error terkecil sebesar 20.54. Penggunaan metode hybrid tersebut juga lebih baik 8.79% daripada regresi linear – ACO dan lebih baik 2.14% dari regresi linear – PSO.

Rahmani, et al. (2013) memprediksi output energi angin jangka pendek menggunakan pemodelan regresi dengan optimasi *hybrid* ACO biasa (bukan kontinyu) dan PSO. *Hybrid* yang dipakai sama dengan yang dipakai oleh Kiran, et al. (2012) berupa perbandingan solusi terbaik yang diperoleh dari ACO dan PSO keduanya per iterasi dan yang terbaik yang disimpan. Model regresi yang dipakai yakni s-curve dan parabola dan nilai *fitness* yang dipakai berupa selisih absolut antara nilai prediksi dan nilai aktual. Data penelitian diperoleh dari Korasan Regional Electric Company (KREC) dengan interval data per jam selama satu tahun mulai dari April 2010 sampai Maret 2011, sedangkan data untuk variabel bebas berupa kecepatan angin dan suhu diperoleh dari Badan meteorology di Nishabour. Hasilnya metode pemodelan regresi linear - hybrid ACO dan PSO memiliki nilai MAPE yang sangat baik yakni 3.513%.

Penelitian lain oleh Huang & Huang (2013) melakukan empat tipe hibridasi antara *Continuous Ant Colony Optimization* (ACO_R) dan PSO untuk diaplikasikan pada permasalahan klustering. Empat tipe hibridasi tersebut yakni sekuensial, paralel, sekuensial dengan memperbesar tabel partikel, dan perbandingan global best. Dari empat tipe hybrid tersebut yang terbaik adalah sekuensial dengan memperbesar tabel partikel karena tabel partikel yang diperbesar menambah diversifikasi (*global exploration*) pada PSO dan ACO_R, efek ini sama dengan memperbesar popSize itu sendiri karena tabel partikel menjadi dua kali lipat lebih besar dari yang awal. Hasil terbaik kedua adalah tipe sekuensial biasa tanpa perlu memperbesar tabel partikel. Dari berbagai macam penelitian-penelitian sebelumnya tersebut dapat dibuat ringkasan yang ditunjukkan pada Tabel 2.1.

Dari penelitian-penelitian sebelumnya tersebut maka pada penelitian ini metode yang digunakan berupa pemodelan regresi linear menggunakan hybrid

PSO dan ACO_R (PSOACO_R) secara sekuensial untuk memprediksi konsumsi energi primer Indonesia. Faktor-faktor yang digunakan untuk menentukan nilai konsumsi energi berupa nilai GDP, GNI, populasi, impor, dan ekspor. Terakhir untuk mengukur seberapa baik metode tersebut digunakan *Mean Average Percentage Error* (MAPE).

Tabel 2.1 Ringkasan penelitian kasus prediksi energi primer sebelumnya

Peneliti	Kasus Prediksi	Metode	Faktor yang digunakan	Tingkat kesalahan
(Yuan, et al., 2017)	Kebutuhan energi primer per provinsi dan nasional di China	Hierarchical bayesian	GDP, populasi, tingkat urbanisasi, konsumsi energi, share secondary industry	RMSE 0.024
(Azadeh, et al., 2013)	Konsumsi energi terbarukan di Iran	Pemodelan regresi linear – multi layer perceptron	GDP, CO ₂ , CO, NO ₂ , harga gas, harga minyak	MAPE 99.9%
(Barak & Sadegh, 2016)	Konsumsi energi primer di Iran	ARIMA-ANFIS	GDP, populasi, impor, ekspor	MSE 0.000263
(Deka, et al., 2016)	Kebutuhan energi primer Amerika serikat	ANN, regresi, dan ARIMA	GDP, GNI, pendapatan per kapita	RMSE 1.99
(Yu, et al., 2012)	Kebutuhan energi primer di China	pemodelan regresi-PSO dan GA	GDP, populasi, proporsi industri, proporsi polulasi urban, prosentase penggunaan batubara	MAPE 0.54%
(Uzlu, et al., 2014)	Konsumsi energi primer di Turki	backpropagation-teaching learning based optimization	GDP, populasi, impor, ekspor	RMSE 2.192
(Kiran, et al., 2012)	Kebutuhan energi primer di Turki	Pemodelan regresi-hybrid ACO dan PSO	GDP, populasi, impor, ekspor	Relative error 20.54
(Rahmani, et al., 2013)	Output energi angin jangka pendek di Iran	Pemodelan regresi-hybrid ACO dan PSO	kecepatan angin, suhu	MAPE 3.513%

2.2 Konsumsi Energi Primer Indonesia

Prediksi merupakan permasalahan penting yang mencakup area luas, diantaranya industri dan bisnis, pemerintahan, ekonomi, ilmu pengetahuan, kesehatan, sosial, politik, dan keuangan (Montgomery, et al., 2008). Permasalahan prediksi berdasarkan jangka waktunya dibagi menjadi tiga yakni jangka pendek, jangka menengah, dan jangka panjang. Prediksi jangka pendek memprediksi periode waktu dalam rentang sedikit berupa beberapa hari, minggu, atau bulan kedepan. Prediksi jangka menengah memprediksi satu sampai dua tahun kedepan, dan prediksi jangka panjang bisa meliputi banyak

tahun atau bahkan dekade. Salah satu penyebab prediksi begitu pentingnya dilakukannya karena prediksi kejadian masa depan merupakan masukan penting untuk perencanaan dan proses penentuan keputusan. Sedangkan menurut tekniknya prediksi terdapat dua macam, kualitatif dan kuantitatif. Teknik prediksi kualitatif digunakan dengan menggunakan bantuan beberapa pakar dan sering digunakan jika sedikit atau bahkan tidak adanya data historis sebagai dasar untuk dilakukannya prediksi. Prediksi kuantitatif menggunakan data historis dan menggunakan sebuah pemodelan. Sebuah model membentuk sebuah pola dalam data dan mengekspresikan sebuah hubungan statistik antara nilai sebelumnya dengan nilai sekarang pada suatu variabel, kemudian model ini digunakan untuk memproyeksikan data masa depan.

Ketidakstabilan politik dan perubahan iklim adalah salah satu faktor eksternal utama yang telah mengubah pandangan tentang energi dan memaksa dunia beradaptasi untuk mencapai keamanan energi jangka panjang yang berkelanjutan (Badan Pengkajian dan Penerapan Teknologi, 2016). SDG (*Sustainable Development Goal*) untuk tahun 2030 yang diresmikan pada United Nations Summit di New York September 2015 lalu merupakan perwujudan usaha dunia internasional dalam menjawab tantangan ini. Penetapan SDG ini mengindikasikan bahwa perencanaan energi yang andal dan berkelanjutan sangat penting untuk mendukung pembangunan jangka panjang. Beberapa tujuan global dalam SDG yang terkait dengan perencanaan energi antara lain energi bersih terjangkau, pertumbuhan ekonomi dan pekerjaan yang layak, dan aksi terhadap perubahan iklim.

Menurut Kementerian Energi dan Sumberdaya Mineral (2016) definisi energi primer adalah energi yang diperoleh dari alam dan belum mengalami proses pengolahan lebih lanjut. Konsumsi energi primer di Indonesia terdiri dari minyak bumi, batubara, gas, dan energi terbarukan (surya, angin, air, biomassa, dan panas bumi). Energi primer umum menggunakan satuan toe (*tonne oil equivalent*) dan jika dikonversikan 1 toe setara dengan 11,63 MWh tenaga listrik, 1,43 ton batubara, 39,68 MBtu gas bumi atau 10.000 MCal.

Faktor yang mempengaruhi konsumsi energi primer menurut Kementerian ESDM (Energi dan Sumberdaya Mineral) berupa satu faktor penentu yakni produk domestik bruto (PDB). Kiran, et al. (2012) menggunakan faktor *gross domestic product* (GDP), populasi penduduk, impor, dan ekspor untuk memprediksi konsumsi energi primer di Turki. Barak & Sadegh (2016) juga menggunakan data GDP, populasi, impor, dan ekspor dari tahun 1967-2012 untuk memprediksi konsumsi energi di Iran. Faktor lain berupa GNI (*gross national income*), GDP, dan populasi dipakai oleh Deka, et al. (2016) untuk memprediksi konsumsi energi di Amerika Serikat. Sehingga faktor-faktor tersebut dipercaya bahwa konsumsi energi primer ditentukan oleh beberapa hal tersebut yakni:

1. GNI (*gross national income*) disebut juga GNP (*gross national product*) dalam beberapa literature.

2. GDP (*gross domestic product*) yang lebih dikenal dengan PDB (produk domestik bruto)
3. Populasi penduduk
4. Nilai impor produk dan jasa
5. Nilai ekspor produk dan jasa

Disisi lain prediksi konsumsi energi primer merupakan hal yang sulit dilakukan, beberapa penyebabnya yakni tingkat pertumbuhan ekonomi yang tinggi, teknologi, keputusan pemerintah, dan ketersediaan data (Barak & Sadegh, 2016; Kiran, et al., 2012; Deka, et al., 2016).

2.3 Regresi Linear Berganda

Pemodelan regresi banyak dipakai untuk memprediksi konsumsi energi primer, beberapa diantaranya, regresi linear (Deka, et al., 2016), regresi linear dan kuadrat (Kiran, et al., 2012), serta regresi linear, eksponensial, dan kuadrat (Yu, et al., 2012). Regresi linear berganda digunakan untuk mengukur pengaruh variabel bebas (X_i) berjumlah lebih dari satu terhadap variabel terikat (Y) (Pascual, et al., 2016). Pemodelan regresi linear berganda ditunjukkan dalam Persamaan 2.1.

$$Y = b_0 + \sum_{i=1}^k b_i X_i \pm \epsilon \quad (2.1)$$

dimana b_0 merupakan intersep. Intersep adalah rata-rata nilai Y saat X_i bernilai 0. b_i merupakan koefisien kemiringan regresi dari variabel bebas X_i , setiap koefisien kemiringan b_i adalah ukuran perubahan pada Y saat X_i diganti dengan nilai 1 dan sisa variabel bebas lain bernilai tetap. Untuk mencari nilai ϵ (*random error*) didapat dari $\epsilon_i = Y_i - \hat{Y}_i$, dimana Y_i adalah nilai aktual ke- i , dan $\hat{Y}_i$ adalah nilai prediksi ke- i .

2.4 Particle Swarm Optimization - PSO

Particle Swarm Optimization (PSO) adalah salah teknik metaheuristik *stochastic* yang diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 untuk menyelesaikan permasalahan kontinyu (Bonyadi, et al., 2014). Dalam PSO, solusi potensial disebut partikel, bergerak dalam ruang pencarian mengikuti partikel optimum sebelumnya dengan meng-*update* posisi partikel setiap iterasinya. Setiap partikel didefinisikan oleh tiga hal yakni:

1. Posisi (x_i). Posisi dari partikel ke- i digunakan untuk mengevaluai kualitas partikel dengan menggunakan nilai *fitness*
2. Kecepatan (v_i). Berfungsi untuk menentukan arah dan besar perpindahan posisi partikel ke- i
3. *Personal best* ($pbest_i$). Pbest adalah posisi terbaik partikel ke- i yang pernah tercapai sepanjang iterasi.

PSO sendiri terinspirasi dari sirkulasi informasi dan perilaku sosial pada kawanan burung dan ikan. Untuk lebih jelasnya, berikut langkah-langkah rinci PSO:

1. Inisialisasi kecepatan partikel dengan nilai 0.
2. Partikel dalam PSO diinisialisasi dengan *uniformly distributed* menggunakan Persamaan 2.2.

$$x_i^d = x_{min}^d + rand(0,1) \cdot (x_{max}^d - x_{min}^d) \quad (2.2)$$

dimana x_{min}^d adalah batas terbawah ruang pencarian dimensi ke- d dan x_{max}^d adalah batas teratas ruang pencarian dimensi ke- d .

3. Inisialisasi pbest dan gbest. Pbest diinisialisasi dengan menyalin semua nilai inisialisasi partikel, sedangkan inisialisasi gbest dicari dengan menemukan pbest dengan nilai *fitness* terbaik
4. Pada tahap iterasi, pertama kecepatan partikel di-*update* menggunakan Persamaan 2.3.

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1 \cdot r_1 \cdot (pbest_i^d(t) - x_i^d(t)) + c_2 \cdot r_2 \cdot (gbest_i^d(t) - x_i^d(t)) \quad (2.3)$$

dimana c_1 adalah *cognitive coefficient* yang berguna untuk menjaga partikel baru yang terbentuk agar cenderung memiliki posisi disekitar pbest iterasi sebelumnya, sedangkan c_2 atau *social coefficient* berguna untuk mempengaruhi partikel baru yang terbentuk agar bergerak mendekati gbest yang ditemukan sebelumnya. Nilai kedua koefisien ini yang sering digunakan adalah 2 atau 2.05 atau 1.496 (Bonyadi, et al., 2014). r_1 dan r_2 merupakan bilangan random *uniformly distributed* yang terletak pada interval (0,1).

Nilai w merupakan parameter inersia yang digunakan untuk mengontrol pengaruh kecepatan iterasi sebelumnya dengan kecepatan sekarang agar tidak berubah secara signifikan, nilai w yang umum digunakan adalah 0.7298 (Bonyadi, et al., 2014). Tingkat konvergensi PSO dipengaruhi oleh 3 nilai tersebut yakni c_1 , c_2 , dan w . Jika ketiga nilai tersebut tidak diatur dengan tepat maka akan menyebabkan kecepatan partikel menjadi tak terbatas, salah satu cara untuk mencegahnya adalah dengan melakukan *velocity clamping* $[-v_{max}, v_{max}]$, dimana v_{max} adalah batas atas kecepatan yang diperbolehkan.

Untuk menentukan v_{max} Marini & Walczak (2015) menggunakan Persamaan 2.4 dimana k adalah nilai konstanta dengan nilai antara 0-1, x_{max}^d adalah batas atas ruang pencarian dimensi ke- d , dan x_{min}^d adalah batas terbawahnya.

$$v_{max}^d = k \frac{(x_{max}^d - x_{min}^d)}{2} \quad (2.4)$$

proses *velocity clamping* sendiri juga bertujuan untuk menyeimbangkan antara *global exploration* dan *local exploitation* PSO. Proses *velocity clamping* ditunjukkan pada Persamaan 2.5.

$$v_i^d(t+1) = \begin{cases} v_{max}^d, & v_i^d(t+1) \geq v_{max}^d \\ -v_{max}^d, & v_i^d(t+1) \leq -v_{max}^d \\ v_i^d(t+1), & \text{lainnya} \end{cases} \quad (2.5)$$

dimana untuk menentukan $v_i^d(t+1)$ atau kecepatan partikel ke- i dimensi ke- d nilai $v_i^d(t+1)$ yang dihasilkan akan diperiksa. Jika nilai $v_i^d(t+1)$ lebih besar dari batas atas v_{max}^d maka nilainya akan diganti oleh v_{max}^d , begitupun sebaliknya jika lebih kecil dari batas bawah $-v_{max}^d$ maka nilainya akan diganti oleh $-v_{max}^d$, jika nilai $v_i^d(t+1)$ berada diantaranya atau sama dengan nilai *velocity clamping* maka nilai $v_i^d(t+1)$ tidak diganti.

5. Update posisi partikel

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (2.6)$$

Untuk meng-update posisi partikel maka nilai setiap dimensi ke- d dari partikel baru ke- i dihitung menggunakan Persamaan 2.6, dimana $x_i^d(t)$ adalah nilai partikel iterasi sebelumnya dan $v_i^d(t+1)$ adalah kecepatan partikel yang saat ini.

$$x_i^d(t+1) = \begin{cases} x_{max}^d, & x_i^d(t+1) \geq x_{max}^d \\ x_{min}^d, & x_i^d(t+1) \leq x_{min}^d \\ x_i^d(t+1), & \text{lainnya} \end{cases} \quad (2.7)$$

Meski sudah melakukan *velocity clamping*, posisi partikel masih bisa keluar dari ruang pencarian maka dilakukanlah *limit* posisi partikel menggunakan Persamaan 2.7, dimana x_{min}^d adalah batas terbawah ruang pencarian dimensi ke- d dan x_{max}^d merupakan batas teratasnya.

6. Update pbest dan gbest. Setiap partikel menyimpan memori dari posisi terbaik sebelumnya dalam pbest yang ditentukan dari baik tidaknya nilai *fitness* (Persamaan 2.8), dan nilai terbaik dari keseluruhan pbest disimpan dalam gbest (Persamaan 2.9). Nilai gbest dicari dengan membandingkan *fitness* pbest terbaik sekarang dengan gbest iterasi sebelumnya. Ulangi langkah 4-6 sampai jumlah iterasi selesai atau minimum konvergen sudah tercapai.

$$pbest_i(t+1) = \begin{cases} x_i(t+1), & f(x_i(t+1)) < f(pbest_i(t)) \\ pbest_i(t), & f(pbest_i(t)) \leq f(x_i(t+1)) \end{cases} \quad (2.8)$$

$$gbest(t+1) = \begin{cases} \min(pbest_i(t+1)), & f(\min(pbest_i(t+1))) < f(gbest(t)) \\ gbest(t), & f(gbest(t)) \leq f(\min(pbest_i(t+1))) \end{cases} \quad (2.9)$$

dimana $pbest_i(t+1)$ adalah pbest iterasi sekarang dan $pbest_i(t)$ merupakan pbest iterasi sebelumnya. $x_i(t+1)$ adalah posisi partikel sekarang dan $gbest(t+1)$ adalah gbest iterasi sekarang.

Langkah-langkah runut metode PSO dapat dijabarkan secara sederhana menggunakan *pseudocode* pada Gambar 2.1.

```

Inisialisasi kecepatan partikel
Inisialisasi partikel (Persamaan 2.2) dan fitnessnya (Persamaan 2.21)
Inisialisasi pbest dan gbest
For iterasi= 1 sampai maxIterasi
    Update kecepatan partikel (Persamaan 2.3)
    Update posisi partikel (Persamaan 2.6) dan hitung fitnessnya
    Update pbest (Persamaan 2.8) dan gbest (Persamaan 2.9)
End For

```

Gambar 2.1 Pseudocode PSO

2.5 Ant Colony Optimization for Continuous Domain - ACO_R

Ant colony optimization (ACO) adalah teknik optimasi probabilistik yang pada awalnya dikembangkan oleh Marco Dorigo untuk menyelesaikan permasalahan optimasi kombinatorial seperti *travelling salesman problem* (TSP), penjadwalan, dan *timetabling* (Socha & Dorigo, 2008). Optimasi kombinatorial bekerja dengan cara menemukan kombinasi atau permutasi terbaik dari komponen permasalahan didalamnya. Akan tetapi dikemudian hari muncul permasalahan optimasi kontinyu yang kurang sesuai dipecahkan dengan ACO. Beberapa diantaranya mungkin bisa diselesaikan dengan cara ACO yakni dengan mengkonversikan nilai pada ruang pencarian menjadi beberapa kumpulan angka. Pendekatan ini sangat menyusahkan, terutama pada kasus ruang pencarian negatif, ruang pencarian yang besar, kasus dengan dimensi permasalahan yang banyak, atau pada kasus yang memerlukan ketelitian tinggi.

Socha & Dorigo (2008) terinspirasi dari ACO mengembangkan metode optimasi baru yang disebut *Ant Colony Optimization for Continuous Domains* (ACO_R) untuk mengatasi permasalahan optimasi kontinyu. Pada penelitian ini *Continuous Ant Colony Optimization* merujuk pada ACO_R yang dikembangkan oleh Socha dan Dorigo ini bukan CACO yang dikembangkan oleh Bilchev dan Parmee pada tahun 1995.

Pada ACO_R kumpulan solusi disimpan dalam sebuah *solution archive* *T*. Pada metode *hybrid* pada penelitian ini tabel solusi disebut tabel partikel, namun tetap mengacu pada *solution archive T* yang sama pada ACO_R. Struktur solusi *archive* ditunjukkan pada Gambar 2.2.

Solusi diurutkan berdasarkan nilai *fitness*-nya, dimulai dari yang terbaik misal untuk permasalahan global minimum maka solusi dengan *fitness* terkecil akan berada paling atas, sehingga nilai $f(s_1) \leq f(s_2) \leq \dots \leq f(s_i) \leq \dots \leq f(s_k)$. Setiap solusi diasosiasikan dengan ω dengan besar nilai $\omega_1 \geq \omega_2 \geq \dots \geq \omega_i \geq \dots \geq \omega_k$. Langkah-langkah ACO_R secara rinci yakni sebagai berikut:

1. Inisialisasi solusi secara random dengan *uniformly distributed* yang ditunjukkan dengan Persamaan 2.10. Persamaan inisialisasi solusi sama dengan inisialisasi partikel pada PSO.

$$s_i^d = s_{min}^d + rand(0,1) \cdot (s_{max}^d - s_{min}^d) \quad (2.10)$$

dimana s_{min}^d adalah nilai batas terbawah ruang pencarian dimensi ke- d , dan s_{max}^d adalah batas teratasnya. Nilai $rand[0,1]$ terdistribusi secara uniform (*uniformly distributed*)

s_1	s_1^1	s_1^2	.	.	.	s_1^d	.	.	.	s_1^D	$f(s_1)$	ω_1
s_2	s_2^1	s_2^2	.	.	.	s_2^d	.	.	.	s_2^D	$f(s_2)$	ω_2
	.	.	.			.	.			.	.	.
	.	.		.		.		.		.	.	.
	.	.			.	.			.	.	.	.
s_i	s_i^1	s_i^2				s_i^d				s_i^D	$f(s_i)$	ω_i
	.	.	.			.	.			.	.	.
	.	.		.		.		.		.	.	.
	.	.			.	.			.	.	.	.
s_g	s_g^1	s_g^2	.	.	.	s_g^d	.	.	.	s_g^D	$f(s_g)$	ω_g
	.	.	.			.	.			.	.	.
	.	.		.		.		.		.	.	.
	.	.			.	.			.	.	.	.
s_k	s_k^1	s_k^2				s_k^d				s_k^D	$f(s_k)$	ω_k

Gambar 2.2 Solusi *archive* pada ACO_R

- Urutkan solusi mulai dari solusi dengan *fitness* terbaik sampai terburuk.
- Hitung ω dengan menggunakan Persamaan 2.11

$$\omega_i = \frac{1}{qk\sqrt{2\pi}} e^{-(i-1)^2/2q^2k^2} \quad (2.11)$$

dimana q merupakan parameter yang mengatur tingkat diversifikasi (tingkat eksplorasi jika pada PSO) dalam ACO_R agar tidak terjebak dalam *local minimum* (Socha & Dorigo, 2008), sedangkan k adalah *archiveSize*. Jika q mempunyai nilai kecil, solusi dengan ranking terbaik lebih dipilih, dan jika bernilai besar maka peluang solusi yang lain terpilih akan semakin merata.

Setelah menghitung ω , hitung peluang p_i (Persamaan 2.12) berdasarkan nilai ω_i yang akan digunakan untuk proses pemilihan partikel menggunakan *roulette wheel* (Huang & Huang, 2013).

$$p_i = \frac{\omega_i}{\sum_{i=1}^k \omega_i} \quad (2.12)$$

- Lakukan perulangan sebanyak m kali untuk membuat semut baru yang berjumlah $m \leq k$ dengan memilih salah satu solusi s_i dengan *roulette wheel*. Setelah solusi s_g terpilih kemudian hitung nilai sigma perdimensi solusi terpilih menggunakan Persamaan 2.13.

$$\sigma_g^d = \xi \sum_{i=1}^k \frac{|s_i^d - s_g^d|}{k-1} \quad (2.13)$$

dimana ξ dengan nilai $\xi > 0$, mempunyai efek yang sama dengan *evaporation rate* ada ACO. Semakin besar nilai ξ , maka tingkat konvergensi akan semakin

melambat (Socha & Dorigo, 2008). s_i^d adalah nilai solusi ke- i dimensi ke- d , s_g^d adalah solusi terpilih dimensi ke- d dan k merupakan *archiveSize*.

Untuk setiap dimensi ke- d semut baru, Fetanat & Khorasaninejad (2015) *generate* random z yang terdistribusi secara normal (*normally distributed*) untuk digunakan pada Persamaan 2.14

$$s_m^d = s_g^d + z \cdot \sigma_g^d \quad (2.14)$$

dimana s_m^d adalah nilai solusi dimensi ke- d semut baru ke- m , σ_g^d merupakan nilai standar deviasi atau sigma dimensi ke- d , dan s_g^d adalah nilai solusi dimensi ke- d yang terpilih menggunakan *roulette wheel*.

Hitung solusi semut baru perdimensi dengan menggunakan Persamaan 2.14 dan limit nilainya agar tidak keluar dari ruang pencarian dengan Persamaan 2.15. Terakhir hitung *fitness* semut baru yang telah dibuat.

$$s_m^d(t+1) = \begin{cases} s_{max}^d, & s_m^d(t+1) \geq s_{max}^d \\ s_{min}^d, & s_m^d(t+1) \leq s_{min}^d \\ s_m^d(t+1), & \text{lainnya} \end{cases} \quad (2.15)$$

5. Gabungkan dan urutkan solusi semut baru dengan solusi sebelumnya.

6. Hapus solusi terburuk agar sesuai ukuran *archiveSize*.

7. *Update* sbest menggunakan Persamaan 2.16

$$sbest(t+1) = s_1(t+1) \quad (2.16)$$

dimana sbest baru adalah set solusi terbaik yakni dengan rangking satu atau teratas. Ulangi langkah 4-7 sampai iterasi selesai atau tingkat *error rate* yang diinginkan. Untuk lebih jelasnya dapat dilihat pseudocode pada Gambar 2.3.

```

Inisialisasi solusi (Persamaan 2.10) dan fitnessnya (Persamaan 2.21)
Urutkan solusi dari yang terbaik
Inisialisasi sbest
Hitung  $\omega$  (Persamaan 2.11) dan  $p$  (Persamaan 2.12)
For iterasi=1 sampai maxIterasi
    For m=1 sampai semutBaru
        Pilih solusi dengan roulette wheel
        For dimensi fitur d=1 sampai D
            Hitung sigma  $\sigma$  (Persamaan 2.13)
            Hitung solusi baru (Persamaan 2.14)
        End For
        Hitung fitness semut baru (Persamaan 2.21)
    End For
    Gabungkan dan urutkan semut baru dengan solusi sebelumnya
    Hapus solusi terburuk agar ukuran sesuai archiveSize
    Update sbest (Persamaan 2.16)
End For

```

Gambar 2.3 Pseudocode ACO_R

2.6 Hybrid Particle Swarm Optimization dan Continuous Ant Colony Optimization - PSOACO_R

Hibridasi terkenal menjadi aspek dasar dari implementasi suatu metode atau algoritma dengan performa yang tinggi. Suatu algoritma yang murni hampir selalu lebih inferior dibandingkan algoritma *hybrid* (Blum, 2005; Huang & Huang, 2013). ACO dan PSO merupakan metode yang paling banyak digunakan untuk menyelesaikan permasalahan optimasi (Marini & Walczak, 2015). *Hybrid* PSO dan ACO_R (PSOACO_R) yang dipakai menggunakan pendekatan *hybrid* sekuensial yang dikembangkan oleh Huang & Huang (2013) yakni proses PSO dan ACO_R dijalankan runut secara bergantian. Pada pendekatan *hybrid* ini PSO dan ACO_R saling berbagi set solusi yang sama yakni dalam tabel partikel.

Pendekatan sekuensial atau jenis *hybrid* 1 dipilih karena memiliki hasil terbaik kedua pada percobaan-percobaan yang dilakukan oleh Huang & Huang (2013). Alasan tidak memilih jenis *hybrid* terbaik yakni jenis *hybrid* 3 berupa pendekatan sekuensial dengan memperbesar tabel partikel karena tabel partikel pada pendekatan *hybrid* 3 diperbesar dua kali lipat pada saat iterasi sehingga hal ini memiliki prinsip yang hampir sama dengan memperbesar ukuran popSize itu sendiri. Tentu *hybrid* dengan popSize lebih besar hasilnya lebih baik daripada popSize setengahnya, peneliti (Huang & Huang) juga memiliki kesimpulan yang sama.

Pada tahap inisialisasi tahap-tahap PSOACO_R sama dengan PSO biasa ditambah perhitungan ω dan peluang p untuk ACO_R. Selama proses iterasi, PSO meng-*update* kecepatan partikel dan *update* posisi partikel berdasarkan tabel partikel.

Inisialisasi kecepatan partikel
Inisialisasi tabel partikel (Persamaan 2.2) dan fitnessnya (Persamaan 2.21)
Inisialisasi pbest dan gbest
Hitung ω (Persamaan 2.11) dan peluang p (Persamaan 2.12)
For iterasi=1 sampai maxIterasi
 Update kecepatan partikel (Persamaan 2.3)
 Update posisi partikel (Persamaan 2.4)
 Update pbest tahap PSO (Persamaan 2.8)
 Update gbest tahap PSO (Persamaan 2.9)
 Buat semut baru (Persamaan 2.13-2.15)
 Replace partikel terburuk (Persamaan 2.17) dan update pbest tahap ACO_R (Persamaan 2.18)
 Update gbest tahap ACO_R (Persamaan 2.19)
End For

Gambar 2.4 Pseudocode PSOACO_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.

$$x_w(ACOR) = \begin{cases} s_m(t+1), & f(s_m(t+1)) < f(x_i(PSO)) \\ x_w(PSO), & f(x_i(PSO)) \leq f(s_m(t+1)) \end{cases} \quad (2.17)$$

dimana $x_w(ACOR)$ adalah proses replace partikel terburuk, $s_m(t+1)$ semut baru ke- m , dan $x_w(PSO)$ adalah partikel dengan *fitness* terburuk pada tahap PSO.

$$pbest_w(ACOR) = \begin{cases} s_m(t+1), & f(s_m(t+1)) < f(pbest_w(PSO)) \\ pbest_w(PSO), & f(pbest_w(PSO)) \leq f(s_m(t+1)) \end{cases} \quad (2.18)$$

$pbest_w(ACOR)$ adalah proses *update* pbest pada tahap PSO, $s_m(t+1)$ semut baru ke- m , dan $pbest_w(PSO)$ adalah pbest yang sejajar dengan partikel yang mempunyai *fitness* terburuk pada tahap PSO.

$$gbest(ACOR) = \begin{cases} s_m(t+1), & f(s_m(t+1)) < f(gbest(PSO)) \\ gbest(PSO), & f(gbest(PSO)) \leq f(s_m(t+1)) \end{cases} \quad (2.19)$$

Terakhir *update* gbest jika *fitness* semut baru lebih baik daripada gbest yang dihasilkan PSO seperti yang ditunjukkan pada Persamaan 2.19. dimana $gbest(ACOR)$ adalah proses *update* gbest tahap ACO_R , $s_m(t+1)$ semut baru ke- m , dan $gbest(PSO)$ adalah gbest yang ditemukan pada iterasi tahap PSO. Agar lebih jelas proses *hybrid* PSOACO_R sekuensial dapat dilihat pseudocode PSOACO_R pada Gambar 2.4.

2.7 Pemodelan Regresi Linear untuk Prediksi Konsumsi Energi Primer Indonesia Linear Menggunakan PSOACO_R

Prediksi energi dikembangkan dari algoritma PSOACO_R untuk menemukan global minimum dari Persamaan 2.21. Lima indikator ekonomi yakni GNI, GDP, populasi, impor, dan ekspor Indonesia digunakan untuk memprediksi konsumsi energi primer Indonesia. Pada penelitian ini prediksi konsumsi energi primer dimodelkan dalam bentuk pemodelan regresi linear berganda seperti pada Persamaan 2.20.

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 \quad (2.20)$$

dimana b_0 adalah bobot konstanta regresi, b_1, b_2, b_3, b_4, b_5 bobot koefisien regresi, dan X_i adalah variabel bebas (GNI, GDP, populasi, impor, dan ekspor). Bobot-bobot tersebut dioptimasi setiap iterasinya dan disimpan dalam gbest dalam bentuk lima dimensi.

PSOACO_R digunakan untuk mengoptimasi pencarian bobot intersep dan kemiringan koefisien regresi. Tujuan utama pencarian bobot ini yakni mencari *fittest model* persamaan regresi dari data latih dengan cara mencari nilai *fitness* terkecil. Nilai *fitness* merupakan tolak ukur seberapa baik set solusi (partikel pada PSO) untuk menyelesaikan suatu permasalahan. Fungsi nilai *fitness* untuk

permasalahan penelitian berupa pencarian global minimum ditunjukkan pada Persamaan 2.21.

$$\min fitness = \sum_{r=1}^R (Y_r - \hat{Y}_r)^2 \quad (2.21)$$

dimana Y_r adalah nilai aktual konsumsi energi primer pada tahun ke- r . $\hat{Y}_r$ adalah nilai prediksi konsumsi energi primer pada tahun ke- r .

2.8 Perhitungan Tingkat Kesalahan

Untuk mengetahui seberapa besar tingkat kesalahan pemodelan regresi linear prediksi konsumsi energi primer yang dihasilkan, maka diperlukan metode untuk menghitung tingkat kesalahan prediksi. Dalam penelitian ini digunakan metode MAPE, metode ini paling tepat digunakan karena tingkat kesalahan dinyatakan dalam skala yang berbeda (Azadeh, et al., 2013). Semakin kecil nilai MAPE maka semakin baik performa algoritma tersebut. Menurut Tsai, et al. (2016) nilai akurasi dikategorikan sangat baik jika MAPE yang dihasilkan kurang dari 10%, dan dikategorikan baik jika MAPE antara 10-20% (Tabel 2.2). Rumus MAPE ditunjukkan pada Persamaan 2.22

$$MAPE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i| / Y_i}{n} \times 100\% \quad (2.22)$$

dimana Y_i adalah nilai aktual, $\hat{Y}_i$ nilai hasil prediksi, dan n adalah jumlah data. Perhitungan prediksi pada penelitian ini dilakukan pada pemodelan regresi linear yang dihasilkan oleh PSOACO_R terhadap data uji dari tahun 2006 sampai 2016.

Tabel 2.2 Kategori akurasi MAPE

MAPE	Kategori akurasi
≤10%	Sangat Baik
10-20%	Baik
20-50%	Cukup
≥50%	Rendah