

BAB 2. TINJAUAN PUSTAKA

Bab tinjauan pustaka pada penelitian ini terdiri dari beberapa sub-bab yang membahas mengenai optimasi, *linear programming*, dan *Firefly Algorithm*. Berikut merupakan penjelasan mengenai sub-subbab tersebut.

2.1 Optimasi dan *Linear Programming*

Konsep, definisi, dan formula mengenai optimasi dan *linear programming* diambil dari Taha (2017). Optimasi merupakan salah satu cabang ilmu matematika yang mempelajari penyelesaian permasalahan untuk mencapai nilai maksimum atau minimum. Berdasarkan definisi tersebut, optimasi diartikan sebagai proses penentuan nilai terbaik (maksimum atau minimum) dari suatu fungsi dengan mempertimbangkan berbagai kendala yang ada. Salah satu pendekatan yang banyak digunakan untuk menyelesaikan permasalahan optimasi adalah *linear programming*. Pendekatan ini menggunakan persamaan dan pertidaksamaan linear untuk memperoleh solusi optimal dengan tetap memperhatikan batasan yang berlaku. Sifat linear menunjukkan bahwa seluruh fungsi dalam model yang digunakan berbentuk linear, yaitu fungsi dengan variabel yang berpangkat satu. Model *linear programming* terdiri dari beberapa unsur utama yaitu :

1. Variabel keputusan

Variabel keputusan adalah variabel-variabel yang nilainya akan mempengaruhi hasil dari fungsi tujuan. Penentuan variabel keputusan harus dilakukan sebelum merumuskan fungsi tujuan dan fungsi kendalanya.

2. Fungsi tujuan

Fungsi tujuan merupakan rumusan matematis yang menjadi sasaran utama dari permasalahan yang dianalisis dalam *linear programming*. Tujuannya dapat berupa memaksimalkan keuntungan atau meminimalkan biaya tergantung pada konteks permasalahan dengan memperhatikan kendala yang ada.

3. Fungsi kendala

Fungsi kendala merupakan batasan atau syarat yang harus dipenuhi dalam nilai variabel keputusan, biasanya dinyatakan dalam bentuk persamaan atau pertidaksamaan sesuai dengan kondisi yang ada.

Model umum *linear programming* secara matematis dibentuk dengan menyusun fungsi tujuan dan fungsi kendala. Fungsi ini diformulasikan seperti pada Persamaan (2.1) sampai (2.3).

Memaksimumkan/minimumkan :

$$Z = \sum_{j=1}^n c_j x_j \quad (2.1)$$

dengan kendala

$$\sum_{j=1}^n a_{ij} x_j \{ \geq, =, \leq \} b_i \quad (2.2)$$

$$x_j \geq 0, \quad (2.3)$$

untuk $i = 1, 2, 3, \dots, m$ dan $j = 1, 2, 3, \dots, n$

dengan

Z : fungsi tujuan yang dicari nilai optimalnya

c_j : koefisien variabel keputusan ke- j

x_j : variabel keputusan atau banyak produk ke- j

a_{ij} : banyaknya sumber ke- i yang dialokasikan oleh setiap unit ke- j

b_i : ketersediaan sumber daya ke- i

m : jumlah kendala yang tersedia

n : jumlah jenis produk yang diproduksi

2.2 Firefly Algorithm

Konsep, definisi, dan formula mengenai *Firefly Algorithm* diambil dari Yang (2010). Pada tahun 2007, Xin-She Yang dari University of Cambridge memperkenalkan sebuah algoritma metaheuristik baru yang dikenal sebagai *Firefly Algorithm*. *Firefly Algorithm* ini terinspirasi dari pola kedipan cahaya yang

dihasilkan oleh kunang-kunang dalam menarik perhatian satu sama lain. Algoritma ini mengikuti tiga prinsip utama, yaitu:

1. Semua *firefly* diasumsikan *unisex* sehingga *firefly* dapat saling tertarik satu sama lain tanpa memperhatikan jenis kelamin *firefly* tersebut.
2. Daya tarik antar *firefly* sebanding dengan intensitas cahaya, dimana *firefly* yang redup akan bergerak menuju *firefly* yang lebih terang, dan jika tidak ada yang lebih terang dalam jangkauannya, maka pergerakan *firefly* terjadi secara acak. Daya tarik tersebut akan menurun seiring bertambahnya jarak.
3. Intensitas cahaya ditentukan oleh nilai fungsi tujuan permasalahan. Pada masalah maksimisasi, intensitas cahaya berbanding lurus dengan nilai fungsi tersebut.

Terdapat beberapa istilah penting yang digunakan dalam *Firefly Algorithm* yaitu sebagai berikut:

1. Intensitas Cahaya

Intensitas cahaya merepresentasikan kualitas suatu solusi yang ditentukan oleh nilai fungsi *fitness*. Semakin baik nilai fungsi *fitness* yang dihasilkan, maka semakin terang cahaya yang dimiliki oleh seekor *firefly*, sehingga *firefly* lain dengan cahaya lebih redup akan bergerak mendekati *firefly* yang lebih terang. Fungsi *fitness* $f(x)$ dalam masalah maksimasi memiliki formula yang sama dengan fungsi tujuan Z dan nilainya berbanding lurus dengan intensitas cahaya $I(x)$. Fungsi tersebut diformulasikan seperti Persamaan (2.4).

$$I(x) = f(x) \quad (2.4)$$

dengan

$I(x)$: Intensitas cahaya

$f(x)$: fungsi *fitness*

2. Daya Tarik

Ketika dua *firefly* berada dalam jarak yang berdekatan, akan muncul gaya tarik-menarik yang dinyatakan dengan simbol beta (β). Daya tarik digunakan untuk mengukur kekuatan suatu *firefly* dalam bergerak menuju *firefly* lainnya.

Besarnya daya tarik tersebut dipengaruhi oleh tingkat kecerahan cahaya yang dipancarkan, sehingga sifatnya relatif dan bergantung pada jarak antar *firefly*. Karena intensitas cahaya yang diterima berbanding lurus dengan kekuatan tarik yang dirasakan, maka nilai daya tarik dapat dirumuskan secara matematis seperti Persamaan (2.5)

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2.5)$$

dengan

β : daya tarik antara dua *firefly* pada jarak r

β_0 : nilai/konstanta keaktraktifan β pada jarak ($r = 0$)

γ : koefisien penyerapan cahaya untuk $\gamma \in [0, \infty]$

r : jarak antara dua *firefly*

3. Jarak

Jarak antara dua *firefly* i dan j dengan posisi x_i dan x_j adalah jarak *euclidean*, dimana selisih koordinat lokasi *firefly* i terhadap *firefly* j merupakan jarak garis lurus terpendek diantara keduanya, yang dirumuskan pada Persamaan (2.6).

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2} \quad (2.6)$$

dengan

r_{ij} : jarak antara *firefly* i dan *firefly* j

x_i : vektor posisi *firefly* i

x_j : vektor posisi *firefly* j

x_i^k : nilai dari vektor x_i pada dimensi ke- k

x_j^k : nilai dari vektor x_j pada dimensi ke- k

k : dimensi dari vektor posisi *firefly*, dengan $k = 1, 2, \dots, d$

d : jumlah dimensi dari vektor posisi *firefly*

4. Pergerakan

Perhitungan jarak selalu melibatkan minimal dua objek, demikian pula pada *Firefly Algorithm* di mana setiap *firefly* membutuhkan *firefly* lain yang lebih

terang untuk dapat bergerak mendekat. Apabila tidak terdapat *firefly* dengan intensitas cahaya lebih tinggi, maka pergerakan dilakukan secara acak. Pergerakan dari *firefly* i yang bergerak ke arah *firefly* j yang lebih terang ditentukan dengan Persamaan (2.7).

$$x_{i_{baru}}^k = x_{i_{lama}}^k + \beta_0 e^{-\gamma r^2} (x_{j_{lama}}^k - x_{i_{lama}}^k) + \alpha \left(rand - \frac{1}{2} \right) \quad (2.7)$$

dengan

$x_{i_{baru}}^k$: nilai dari vektor posisi baru *firefly* i pada dimensi ke- k

$x_{i_{lama}}^k$: nilai dari vektor posisi lama *firefly* i pada dimensi ke- k

$\beta_0 e^{-\gamma r^2}$: daya tarik (*attractiveness*) antara dua *firefly* pada jarak r

$x_{j_{lama}}^k$: nilai dari vektor posisi lama *firefly* j pada dimensi ke- k

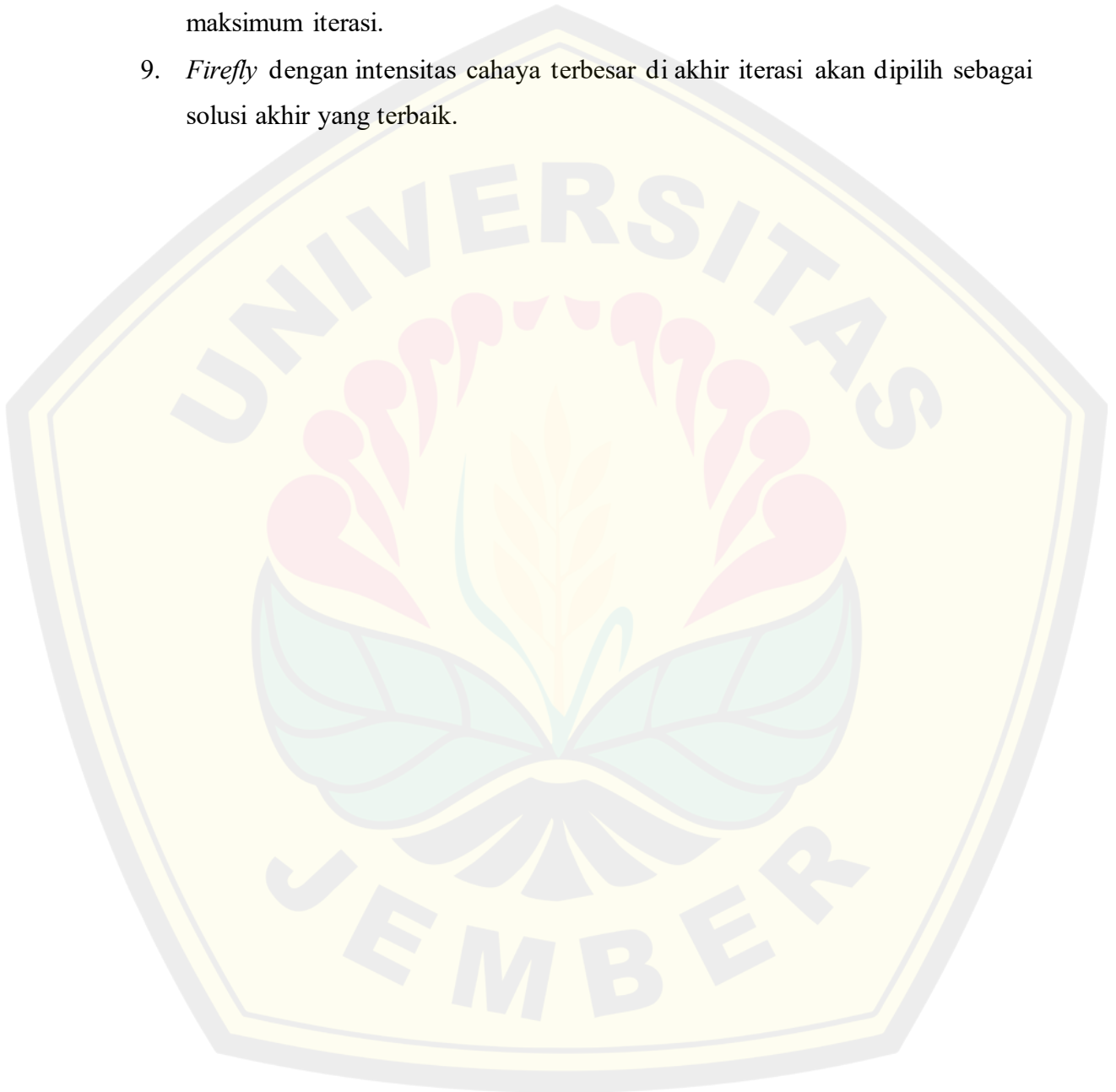
α : koefisien parameter acak

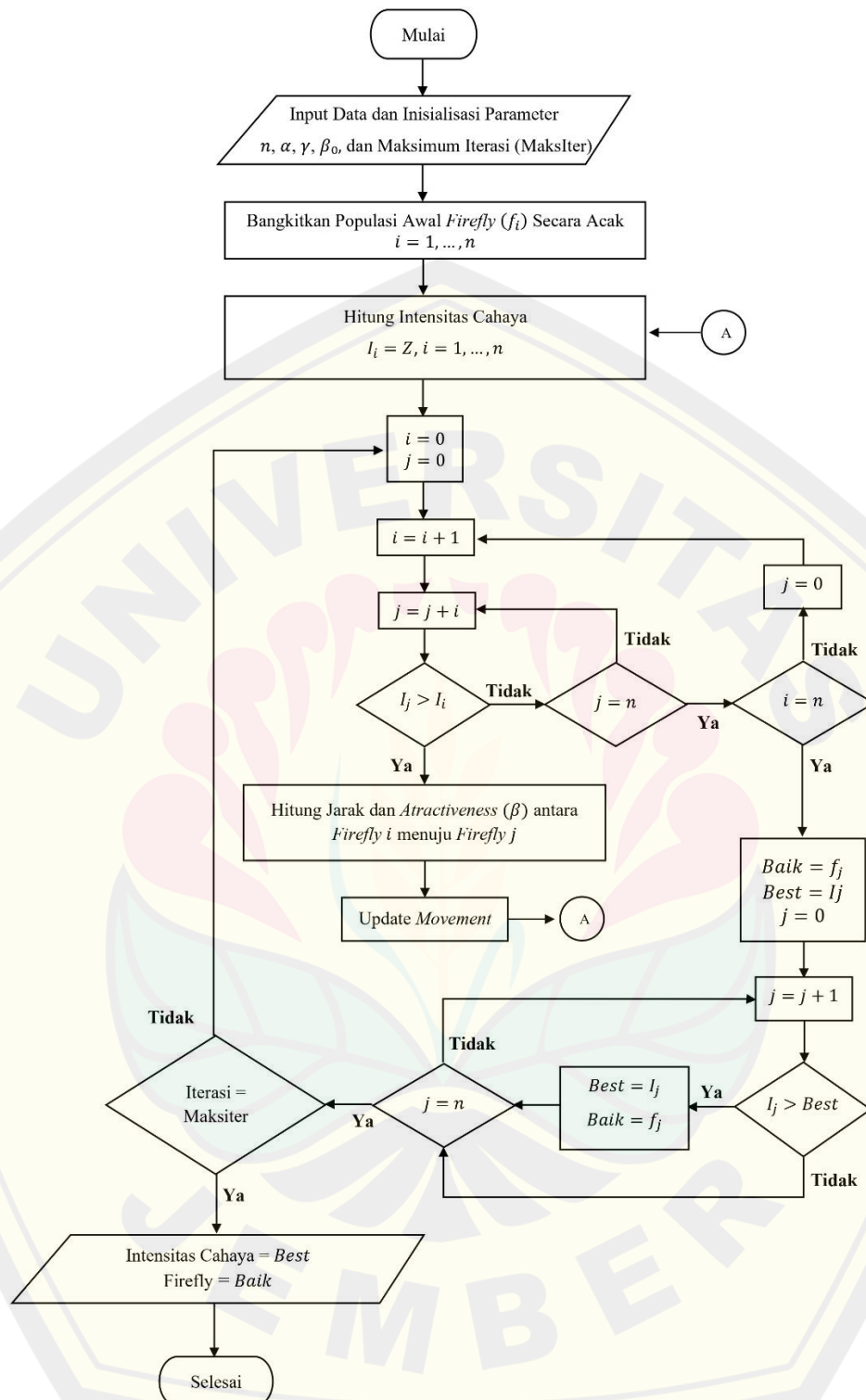
$rand$: bilangan real acak pada interval $[0,1]$

Diagram alir mengenai tahapan metode *Firefly Algorithm* dalam menyelesaikan permasalahan optimasi ditunjukkan pada Gambar 2.1 dengan penjelasan sebagai berikut:

1. Tahapan dimulai dengan input data dan inisialisasi parameter awal yaitu jumlah *firefly* (n), koefisien parameter acak (α), koefisien penyerapan cahaya (γ), konstanta keaktraktifan (β_0), dan maksimum iterasi.
2. Bangkitkan populasi awal sebanyak n *firefly* secara acak.
3. Hitung intensitas cahaya setiap *firefly* berdasarkan fungsi *fitness* menggunakan Persamaan (2.4) dan simpan nilai intensitas cahaya terbaik.
4. Proses iterasi dimulai dari membandingkan *firefly* j dengan *firefly* i . Jika intensitas cahaya *firefly* j lebih besar dari *firefly* i maka dua *firefly* ini kemudian dihitung nilai jarak dan daya tarik menggunakan Persamaan (2.6) dan (2.5).
5. *Update* posisi *firefly* i menuju posisi *firefly* j yang lebih terang menggunakan Persamaan (2.7).

6. Setelah semua posisi *firefly* diperbarui, hitung intensitas cahaya masing masing *firefly* pada lokasi barunya dan tentukan solusi terbaik *firefly* dengan intensitas cahaya terbesar.
7. Solusi terbaik akan disimpan untuk iterasi selanjutnya.
8. Ulangi proses pada langkah 3 sampai langkah 7 hingga mencapai batas maksimum iterasi.
9. *Firefly* dengan intensitas cahaya terbesar di akhir iterasi akan dipilih sebagai solusi akhir yang terbaik.





Gambar 2.1 Diagram Alir