

# Optimasi Fungsi Multimodal Menggunakan *Flower Pollination Algorithm* Dengan Teknik *Clustering*

## *Multimodal Function Optimization Using Flower Pollination Algorithm With Clustering*

Rahmat Karim<sup>1</sup>, Kuntjoro Adji Sidarto<sup>2</sup>, Suharsono Bantun<sup>3</sup>

<sup>1,3</sup>Program Studi Sistem Informasi, Universitas Sembilanbelas November Kolaka

<sup>2</sup>Departemen Matematika, Institut Teknologi Bandung

E-mail: <sup>1</sup>rahmatkarim@usn.ac.id, <sup>2</sup>sidarto@math.itb.ac.id, <sup>3</sup>suharsonob@usn.ac.id

### Abstrak

Optimasi fungsi multimodal merupakan permasalahan yang banyak dijumpai dalam bidang teknik, sains, ilmu sosial dan ekonomi. Tujuan utama dari permasalahan multimodal adalah untuk melokalisasi semua solusi yang tersedia baik optimum lokal maupun optimum global dalam sekali *running*. *Flower Pollination Algorithm* yang umum digunakan untuk optimasi global perlu dimodifikasi dan dikembangkan agar dapat menyelesaikan tantangan dalam optimasi fungsi multimodal. Pada penelitian ini kami mengkombinasikan *Flower Pollination Algorithm* dengan teknik *Clustering* untuk mengoptimasi fungsi multimodal. Dalam uji coba terhadap 5 fungsi *benchmark* multimodal yaitu *Second minima*, *Six hump camel back*, *Rastrigin*, *Vincent* dan *Shubert* diperoleh hasil bahwa metode yang diusulkan (FPAC) sukses menemukan semua solusi dari masing-masing fungsi multimodal dalam sekali *running* baik untuk kasus dimensi rendah maupun dimensi tinggi.

Kata kunci: Optimasi fungsi multimodal, *Flower Pollination Algorithm*, *Clustering*, FPAC

### Abstract

*Multimodal function optimization is a problem that is often found in the fields of engineering, science, social sciences and economics. The main objective of multimodal problems is to localize all available solutions both local and global optimum in a single run. Flower Pollination Algorithms that are commonly used for global optimization need to be modified in order to solve the challenges of multimodal optimization. In this study we combined the Flower Pollination Algorithm with Clustering techniques to optimize multimodal functions. The results of 5 benchmark functions of multimodal, namely Second minima, Six hump camel back, Rastrigin, Vincent and Shubert showed that the proposed method (FPAC) succeeded in finding all solutions of each multimodal function in a single run both low and high dimensional cases.*

*Keywords: Multimodal optimization, Flower Pollination Algorithm, Clustering, FPAC*

## 1. PENDAHULUAN

Optimasi fungsi multimodal merupakan permasalahan optimasi yang direpresentasikan oleh fungsi yang tidak hanya memiliki satu optimum lokal atau global, tapi juga memiliki beberapa optimum lokal dan global. Permasalahan optimasi seperti ini banyak ditemukan di bidang teknik, sains, ilmu sosial dan ekonomi [1]. Tujuan utama dari optimasi multimodal adalah untuk menemukan semua solusi optimum lokal dan global dalam sekali *running*. Dengan mengetahui semua solusi baik optimum lokal maupun global maka para pembuat kebijakan memiliki sejumlah pilihan solusi untuk menyelesaikan permasalahan yang mereka hadapi [2]. Sebagai ilustrasi, gas dapat membeku saat proses distribusi pada sebuah pipa dan para pembuat keputusan dapat melakukan intervensi dengan melakukan rekayasa suhu pipa pada setiap titik tertentu jika mengetahui bahwa pada titik-titik itu gas akan segera mengalami penurunan suhu.

Pada masalah optimasi global, algoritma klasik berbasis gradien telah umum digunakan. Akan tetapi pada masalah optimasi multimodal, algoritma tersebut tidak direkomendasikan untuk digunakan. Algoritma berbasis gradien umumnya tidak dapat bekerja dengan baik ketika dihadapkan pada fungsi objektif yang tidak terdeferensial, memiliki gradien yang berfluktuatif, dan tidak memiliki tebakan awal [3]. Sebagai alternatif, algoritma evolusioner yang berbasis stokastik seperti *Genetic Algorithm* (GA), *Differential Evolution* (DE), *Particle Swarm optimization* (PSO) dan *Flower Pollination Algorithm* (FPA) memiliki potensi yang besar untuk menyelesaikan permasalahan multimodal. Algoritma-algoritma evolusioner umumnya dicirikan dengan sifat ekstensifikasi dan intensifikasi yang menjamin algoritma-algoritma evolusioner untuk selalu konvergen pada setiap daerah pencarian [4].

Meskipun ampuh, algoritma-algoritma evolusioner tidak dapat langsung diterapkan pada kasus optimasi fungsi multimodal. Algoritma-algoritma tersebut akan selalu terjebak pada optimum global sehingga dibutuhkan *running* berulang kali untuk melokalisir solusi-solusi lain yang ada dalam daerah pencarian fungsi objektif dan jelas hal itu tidak efektif. Oleh karena itu, algoritma-algoritma evolusioner perlu dimodifikasi dan dikembangkan untuk bisa menemukan semua solusi pada fungsi multimodal dalam sekali *running*. Dalam [3], penulis memodifikasi *species-based* yang merupakan salah satu teknik *Niching* yang mendukung pembentukan sub-sub populasi dalam daerah pencarian dan pada setiap sub populasi yang terbentuk dioptimasi menggunakan algoritma *Differential Evolution*. Modifikasi *species-based differential evolution* dilaporkan memiliki kapasitas untuk menemukan beberapa optimum lokal dan global dalam pada permasalahan multimodal.

Dalam [5], penulis memodifikasi algoritma *Firefly* dengan *K-means clustering* dan *simulated annealing* untuk menyelesaikan permasalahan multimodal. *K-means clustering* digunakan untuk membagi populasi ke dalam sub-sub populasi. Stabilitas kriteria kemudian digunakan untuk menentukan sub-sub populasi yang stabil dan tidak stabil. Sementara, *Simulated annealing* digunakan untuk meningkatkan kekuatan pencarian, akurasi dan kecepatan algoritma. Kombinasi ketiga metode tersebut dilaporkan memiliki kemampuan tinggi dalam menemukan solusi-solusi optimal dari fungsi multimodal.

Dalam [6], penulis memodifikasi *Flower Pollination Algorithm* dengan mengkombinasikan 3 elemen yaitu *Memory Mechanism* (MM), *Selection Strategy* (SS), dan *Depuration Procedure* (DP). *Multimodal Flower Pollination Algorithm* kemudian diuji menggunakan 14 fungsi *benchmark* dan performanya dibandingkan dengan 4 algoritma multimodal yaitu CDE, FSDE, DCGA, dan CSA. Hasil eksperimen mengindikasikan bahwa MFPA lebih baik dari algoritma-algoritma lain yang sudah dikenal dalam menyelesaikan permasalahan multimodal.

Dalam [7], BA Model dan algoritma *PageRank* digunakan dalam memodifikasi algoritma *Particle Swarm Optimization* untuk dapat menyelesaikan permasalahan multimodal dan disebut dengan nama MPSOPR. Dalam MPSOPR, setiap partikel tidak hanya belajar dari posisi terbaik tapi juga belajar dari tetangganya dimana setiap tetangganya dipertimbangkan berdasarkan nilai yang dikalkulasi menggunakan algoritma *PageRank*. Performa MPSOPR divalidasi dengan menggunakan 17 fungsi *benchmark* yang dibagi dalam 3 kelompok berdasarkan tingkat kesulitan dan performanya dibandingkan dengan 9 varian dari PSO. MPSOPR secara signifikan membantu meningkatkan PSO dalam menyelesaikan permasalahan multimodal.

Masalah utama yang dihadapi oleh algoritma-algoritma evolusioner ketika digabungkan dengan metode-metode *Niching* untuk menyelesaikan permasalahan multimodal adalah terletak pada penentuan pusat *cluster*. *Cluster* yang terbentuk pada daerah pencarian umumnya tidak potensial sehingga ketika dioptimasi, algoritma evolusioner konvergen pada sesuatu yang bukan solusi dalam permasalahan multimodal. Oleh karena itu, dalam penelitian ini kami mengusulkan penggunaan teknik *Clustering* pada salah satu algoritma evolusioner yaitu *Flower Pollination Algorithm* (FPA) dengan tujuan untuk menemukan semua solusi baik optimum lokal ataupun global dari permasalahan multimodal dalam sekali *running*. Teknik *clustering* pada awalnya dikembangkan oleh Sidarto untuk menyelesaikan sistem persamaan nonlinear [8]. Sementara

FPA dikembangkan oleh Yang untuk menyelesaikan permasalahan optimasi global [9]. Modifikasi dari kedua teknik tersebut terbagi atas 3 fase yaitu fase pertama adalah diversifikasi dimana teknik *Clustering* digunakan untuk menyeleksi apakah vektor-vektor yang disebar dalam daerah pencarian layak dijadikan sebagai pusat *cluster* potensial. Fase kedua adalah intensifikasi dimana FPA digunakan untuk mengoptimasi setiap *cluster* yang terbentuk. Pada fase ketiga dilakukan proses Elitis untuk menyeleksi solusi akhir dari permasalahan multimodal. Modifikasi *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC) ini divalidasi dengan menggunakan 5 fungsi *benchmark* multimodal yang dikompetisikan pada ajang CEC [10]. Hasil eksperimen menunjukkan bahwa metode yang kami usulkan yaitu FPAC sukses menemukan semua solusi pada masing-masing fungsi *benchmark* multimodal dalam sekali *running*.

## 2. METODE PENELITIAN

### 2.1 Karakteristik Flower Pollination Algorithm

Pada tahun 2012, Xin-She Yang mengembangkan algoritma untuk menyelesaikan kasus optimasi global yang terinspirasi dari proses penyerbukan bunga [9]. Algoritma ini dikenal dengan nama *Flower Pollination Algorithm (FPA)*. Asumsi yang digunakan pada metode *Flower Pollination Algorithm* adalah:

1. *Biotic* dan *cross pollination* merupakan proses penyerbukan global dan polinator pada penyerbukan global ini mengikuti gerak *Levy flight*.
2. *Abiotic* dan *self pollination* merupakan penyerbukan lokal.
3. Pollinator seperti serangga dapat menemukan bunga dengan spesies yang sama dan peluang reproduksi proporsional terjadi ketika bunga berasal dari satu spesies.
4. Interaksi atau perubahan dari penyerbukan lokal ke penyerbukan global dapat dikontrol menggunakan switch probability  $p \in [0,1]$ .

Ada dua hal penting dalam algoritma FPA yaitu pergerakan serbuk sari yang dapat terjadi secara lokal atau global. Pada penyerbukan global, serbuk sari dibawa oleh pollinator seperti serangga dan serbuk sari bergerak cukup jauh. Karena serangga sering terbang dan berpindah dalam jarak jauh, penyerbukan global ini diasumsikan mengikuti gerak *Levy flight*. *Levy flight* merupakan gerak acak yang berdistribusi *Levy* dan diadaptasi dari konsep pencarian makanan secara acak pada hewan ketika terbang. Dalam [11], Mantegna mendefinisikan *step size* untuk *Levy flight* adalah sebagai berikut:

$$s = \frac{U}{|V|^{\frac{1}{\lambda}}} \quad (1)$$

Pada FPA penyerbukan secara global, dapat dituliskan secara matematis sebagai berikut:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma \mathbf{L}(\lambda)(\mathbf{g}_* - \mathbf{x}_i^t) \quad (2)$$

Dimana :

$\mathbf{x}_i^t$  adalah serbuk sari ke  $i$  atau vektor solusi  $\mathbf{x}_i$  pada iterasi ke  $t$ .

$\mathbf{g}_*$  adalah solusi terbaik sementara.

$\mathbf{L}$  adalah *step size* dengan *Levy flight* yang berdimensi  $d$ .

$\gamma$  adalah faktor skala untuk *step size*.

$\lambda$  adalah parameter *Levy flight* ( $1 \leq \lambda \leq 2$ ).

Untuk penyerbukan lokal, secara matematis dapat ditulis sebagai berikut:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon(\mathbf{x}_j^t - \mathbf{x}_k^t) \quad (3)$$

Dimana  $\mathbf{x}_j^t$  dan  $\mathbf{x}_k^t$  merupakan serbuk sari dari bunga yang berbeda namun masih dalam satu spesies. Secara matematis,  $\mathbf{x}_j^t$  dan  $\mathbf{x}_k^t$  berasal dari spesies yang sama maka pergerakan diatas ekuivalen dengan *random walks* lokal jika  $\epsilon \sim U(0,1)$  [9].

## 2.2 Flower Pollination Algorithm

Misalkan terdapat masalah optimasi dalam bentuk:

$$\text{minimize } F(\mathbf{x}) \quad (4)$$

dengan  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ , maka pada gambar 1 adalah *Pseudocode* dari FPA [9]:

<ul style="list-style-type: none"> <li>• <b>Masukan parameter</b> yang dibutuhkan Serbuk sari (<math>m</math>), dimensi (<math>d</math>), faktor skala (<math>\gamma</math>), parameter <i>Levy</i> (<math>\lambda</math>), parameter <i>switch</i> (<math>\rho</math>), batas bawah (<math>\mathbf{Bb}</math>), batas atas (<math>\mathbf{Ba}</math>), &amp; iterasi maksimum (<math>k_{maks}</math>).</li> <li>• <b>Inisiasi awal</b> untuk <math>m</math> serbuk / benih bunga <math>\mathbf{x}_i^0 = \mathbf{Bb} + (\mathbf{Ba} - \mathbf{Bb}) \cdot \text{rand}(1, d)</math> untuk <math>i = 1, 2, \dots, m</math> Hitung <math>F(\mathbf{x}_i^0)</math> untuk <math>i = 1, 2, \dots, m</math></li> <li>• <b>Simpan serbuk terbaik</b> sementara <math>\mathbf{g}_* = \mathbf{x}_{i_g}^0, i_g = \text{arg}_i \min (F(\mathbf{x}_i^0))</math> untuk <math>i = 1, 2, \dots, m</math></li> <li>• <b>While</b> <math>t \leq k_{maks}</math> <ul style="list-style-type: none"> <li><b>Lakukan perpindahan serbuk sari</b></li> <li><b>for</b> <math>i = 1:m</math> <ul style="list-style-type: none"> <li><b>if</b> <math>\text{rand} &lt; p</math> <ul style="list-style-type: none"> <li>Hitung vektor <math>\mathbf{L}</math> yang berdimensi <math>d</math> dengan <i>Levy flight</i></li> <li><math display="block">\mathbf{L}(\lambda) = \frac{\text{randn}(1,d) \times \left[ \frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \times \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right]^{1/\lambda}}{ \text{randn}(1,d) ^{1/\lambda}}</math></li> <li>Hitung <math>\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma \mathbf{L}(\lambda)(\mathbf{g}_* - \mathbf{x}_i^t)</math></li> </ul> </li> <li><b>else</b> <ul style="list-style-type: none"> <li>Hitung <math>\epsilon \sim U[0,1]</math></li> <li>Hitung <math>\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon(\mathbf{x}_j^t - \mathbf{x}_k^t)</math></li> </ul> </li> <li><b>end if</b></li> <li>Hitung <math>f(\mathbf{x}_i^{t+1})</math></li> <li>Evaluasi solusi <math>f</math> pada masing-masing serbuk sari</li> <li>Jika solusi <math>\mathbf{x}_i^{t+1}</math> lebih baik maka serbuk sari pindah</li> </ul> </li> <li><b>end for</b></li> <li><b>Update</b> <math>\mathbf{g}_*</math> dan <math>f_{min}</math></li> </ul> </li> <li><b>end while</b></li> </ul>
---

Gambar 1 Flower Pollination Algorithm (FPA)

## 2.3 Teknik Clustering

Secara umum, permasalahan optimasi fungsi multimodal memiliki sejumlah optimum lokal dan optimum global. Algoritma FPA pada gambar 1 hanya akan memperoleh satu solusi yaitu optimum global. Untuk melokalisir semua solusi dari fungsi multimodal dalam sekali *running* maka teknik *Clustering* dibutuhkan. Klusterisasi (*clustering*) umumnya didefinisikan sebagai suatu proses mengelompokkan obyek dengan memaksimalkan kesamaan antar anggota dalam satu *cluster* dan meminimalkan kesamaan antar satu *cluster* dengan cluster lainnya [12]. *Cluster* dengan pusat  $\mathbf{x}$  dan jarak  $\rho$  dalam penelitian ini adalah semua vektor-vektor  $\mathbf{y}$  yang memenuhi  $|\mathbf{x} - \mathbf{y}| < \rho$ . Berikut adalah teknik *Clustering* yang dikembangkan oleh Sidarto [8]:

**Input** : jumlah titik ( $m_{cl}$ ), batas bawah ( $Bb$ ), batas atas ( $Ba$ )

**Proses** :

1. Bangkitkan  $m_{cl}$  sebagai vektor awal  $\mathbf{x}_i(0) \in \mathbb{R}^n$ ,  $i = 1, 2, \dots, m_{cl}$  pada daerah pencarian  $D = [Bb_1, Ba_1] \times [Bb_2, Ba_2] \cdots [Bb_n, Ba_n]$

2. Atur  $\mathbf{x}^* = \mathbf{x}_{i_g}(0)$ ,  $i_g = \arg \min F(\mathbf{x}_i(0))$ ,  $i = 1, 2, \dots, m_{cl}$  sebagai pusat dari *cluster* awal dengan jari-jari *cluster*  $r = \frac{1}{2}(\min_l |Ba_l - Bb_l|)$ ,  $l = 1, 2, \dots, n$ .
3. Untuk setiap  $i = 1, 2, \dots, m_{cl}$  lakukan:  
 Jika  $\mathbf{x}_i$  adalah bukan pusat dari *cluster* yang ada, maka lakukan fungsi *cluster* berikut dengan  $\mathbf{x}_i$  sebagai input. Pada fungsi *cluster* (input  $\mathbf{y}$ ).
  - a. Temukan sebuah *cluster* yang pusat *cluster* nya dekat dengan  $\mathbf{y}$ . Misalkan  $C$  adalah *cluster* tersebut dengan pusat *cluster*  $\mathbf{x}_c$ .
  - b. Atur  $\mathbf{x}_m$  sebagai titik tengah diantara  $\mathbf{y}$  dan  $\mathbf{x}_c$ .
  - c. Bandingkan  $F(\mathbf{y})$ ,  $F(\mathbf{x}_c)$ , dan  $F(\mathbf{x}_m)$ :
    - o Jika  $F(\mathbf{x}_m) < F(\mathbf{y})$  dan  $F(\mathbf{x}_m) < F(\mathbf{x}_c)$ , atur sebuah *cluster* baru dengan pusat  $\mathbf{y}$  dan jari-jari  $\rho = \|\mathbf{y} - \mathbf{x}_m\|$ .
    - o Jika  $F(\mathbf{x}_m) > F(\mathbf{y})$  dan  $F(\mathbf{x}_m) > F(\mathbf{x}_c)$ , atur sebuah *cluster* baru dengan pusat  $\mathbf{y}$  dan jari-jari  $\rho = \|\mathbf{y} - \mathbf{x}_m\|$ .
    - o Jika  $F(\mathbf{y}) > F(\mathbf{x}_c)$ , atur  $\mathbf{y}$  sebagai pusat *cluster* dari  $C$
  - d. Ganti jari-jari dari  $C$  sama dengan jarak diantara  $\mathbf{y}$  dan  $\mathbf{x}_m$

**Output:**  $\mathbf{x}_{c_i} = (\mathbf{x}_{1,i}, \mathbf{x}_{2,i}, \dots, \mathbf{x}_{n,i})^T$  adalah pusat-pusat *cluster* dengan jari-jari  $\rho_i$  yang adaptif untuk  $i = 1, 2, \dots, n$

#### 2.4 Flower Pollination Algorithm dengan Teknik Clustering (FPAC)

*Flower Pollination Algorithm* merupakan algoritma yang akan selalu konvergen pada optimum global, sementara teknik *Clustering* menyediakan *cluster-cluster* potensial dengan jari-jari yang adaptif. Kombinasi kedua teknik tersebut diharapkan dapat melokalisir semua solusi dari fungsi multimodal baik optimum lokal maupun global dalam sekali *running*.

Metode yang diusulkan yaitu *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC), terdiri atas 3 tahap yaitu:

1. Fase diversifikasi.  
 Pada fase ini teknik *Clustering* digunakan dengan tujuan untuk menghasilkan pusat-pusat *cluster* yang potensial pada daerah pencarian fungsi objektif multimodal. Setiap pusat *cluster* yang ditemukan memiliki jari-jari  $\rho$  yang adaptif.
2. Fase intensifikasi  
 Pada fase ini *cluster-cluster* dibentuk. Luasan setiap *cluster* dalam daerah pencarian bergantung pada jari-jari masing-masing *cluster*. Pada setiap *cluster* yang terbentuk kemudian dilakukang proses optimasi menggunakan *FPA* sehingga menghasilkan kandidat solusi sebanyak *cluster* yang terbentuk
3. Fase elitis  
 Pada fase ini solusi akhir dipilih sehingga tidak ada solusi yang bertumpuk pada titik yang sama dalam daerah pencarian fungsi objektif multimodal.

Untuk lebih detail berikut adalah metode FPAC untuk mengoptimasi fungsi multimodal:

**Input:**  $m_{cl}, \gamma_{cl}, \lambda_{cl}, p_{cl}, k_{cl}$  parameter untuk FPA pada fase diversifikasi.  
 $\varepsilon (0 < \varepsilon < 1)$  parameter untuk kandidat solusi optimum.  
 $\delta (0 < \delta < 1)$  parameter untuk membedakan satu kandidat dengan kandidat solusi lainnya jika jarak diantaranya cukup.  
 $m, \gamma, \lambda, p, k_{max}$  parameter untuk FPA pada fase intensifikas

**Proses :**

- **Fase Diversifikasi**

Langkah 1: Bangkitkan  $m_{cl}$  sebagai vektor awal  $\mathbf{x}_i(0) \in \mathbb{R}^n$ ,  $i = 1, 2, \dots, m_{cl}$  pada daerah pencarian  $D = [Bb_1, Ba_1] \times [Bb_2, Ba_2] \dots [Bb_n, Ba_n]$  dan atur  $k = 0$ .

- Langkah 2: Atur  $\mathbf{x}^* = \mathbf{x}_{i_g}(0)$ ,  $i_g = \arg \min F(\mathbf{x}_i(0))$ ,  $i = 1, 2, \dots, m_{cl}$  sebagai pusat dari *cluster* awal dengan jari-jari *cluster*  $r = \frac{1}{2}(\min_l |Ba_l - Bb_l|)$ ,  $l = 1, 2, \dots, n$ .
- Langkah 3: Untuk setiap  $i = 1, 2, \dots, m_{cl}$  lakukan:  
 Jika  $\mathbf{x}_i$  adalah bukan pusat dari *cluster* yang ada, maka lakukan fungsi *cluster* berikut dengan  $\mathbf{x}_i$  sebagai input. Pada fungsi *cluster* (input  $\mathbf{y}$ ).
- Temukan sebuah *cluster* yang pusat *cluster* nya dekat dengan  $\mathbf{y}$ . Misalkan  $C$  adalah *cluster* tersebut dengan pusat *cluster*  $\mathbf{x}_c$ .
  - Atur  $\mathbf{x}_m$  sebagai titik tengah antara  $\mathbf{y}$  dan  $\mathbf{x}_c$ .
  - Bandingkan  $F(\mathbf{y})$ ,  $F(\mathbf{x}_c)$ , dan  $F(\mathbf{x}_m)$ :
    - Jika  $F(\mathbf{x}_m) < F(\mathbf{y})$  dan  $F(\mathbf{x}_m) < F(\mathbf{x}_c)$ , atur sebuah *cluster* baru dengan pusat  $\mathbf{y}$  dan jari-jari  $\rho = \|\mathbf{y} - \mathbf{x}_m\|$ .
    - Jika  $F(\mathbf{x}_m) > F(\mathbf{y})$  dan  $F(\mathbf{x}_m) > F(\mathbf{x}_c)$ , atur sebuah *cluster* baru dengan pusat  $\mathbf{y}$  dan jari-jari  $\rho = \|\mathbf{y} - \mathbf{x}_m\|$ .
    - Jika  $F(\mathbf{y}) > F(\mathbf{x}_c)$ , atur  $\mathbf{y}$  sebagai pusat *cluster* dari  $C$ .
  - Ganti jari-jari dari  $C$  sama dengan jarak diantara  $\mathbf{y}$  dan  $\mathbf{x}_m$ .
- Langkah 4: Lakukan proses perpindahan serbuk pada FPA untuk mengeksplorasi kandidat solusi lain dalam daerah pencarian.
- for**  $i = 1: m_{cl}$   
   **if**  $rand < p$   
     ○ Hitung vektor  $\mathbf{L}$  yang berdimensi  $d$  dengan *Levy flight*  
       
$$\mathbf{L}(\lambda) = \frac{randn(1,d) \times \left[ \frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \times \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right]^{1/\lambda}}{|randn(1,d)|^{1/\lambda}}$$
  
     ○ Hitung  $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \gamma \mathbf{L}(\lambda)(\mathbf{g}_* - \mathbf{x}_i^t)$   
   **else**  
     ○ Hitung  $\epsilon \sim U[0,1]$   
     ○ Hitung  $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon(\mathbf{x}_j^t - \mathbf{x}_k^t)$   
   **end if**  
**end for**
- Langkah 5: Lakukan langkah 3 sampai langkah 5 sebanyak  $k_{cl}$  kali.  
 - **Fase Intensifikasi**
- Langkah 6: Setelah selesai pada fase Diversifikasi kita menghasilkan sejumlah *cluster*:  $C_1, C_2, \dots, C_{n_k}$ . Setiap *cluster* memiliki pusat  $\mathbf{x}_{C_i}$  dan jari-jari  $\rho_i$  yang adaptif, ( $i = 1, 2, \dots, n_c$ ). Pada setiap *cluster* dilakukan FPA untuk menghasilkan kandidat solusi optimum. Pada fase ini digunakan  $m, \gamma, \lambda, p, k_{max}$  sebagai input FPA, dengan  $D_i = [x_{1,i} - \rho_i, x_{1,i} + \rho_i] + [x_{2,i} - \rho_i, x_{2,i} + \rho_i] + \dots + [x_{n,i} - \rho_i, x_{n,i} + \rho_i] \subset \mathbb{R}^n$  sebagai domain dari daerah pencarian dimana  $\mathbf{x}_{C_i} = (\mathbf{x}_{1,i}, \mathbf{x}_{2,i}, \dots, \mathbf{x}_{n,i})^T$  adalah pusat *cluster* untuk  $i = 1, 2, \dots, n$ .
- **Fase Elitis**
- Langkah 7: Simpan kandidat solusi optimum yang hanya memenuhi kondisi berikut:  $F(\mathbf{x} - \epsilon) < F(\mathbf{x})$  dan  $F(\mathbf{x} + \epsilon) < F(\mathbf{x})$ .
- Langkah 8: Misalkan dari langkah 7 diperoleh  $n_g$  kandidat solusi optimum. Dari  $n_g$  kandidat, akan dipilih solusi akhir yang memenuhi  $\|\mathbf{x}_i - \mathbf{x}_j\| > \delta$  untuk  $i, j = 1, 2, \dots, n$  dimana  $\|\mathbf{x}_i - \mathbf{x}_j\|$  adalah jarak diantara  $\mathbf{x}_i$  dan  $\mathbf{x}_j$ . Untuk kasus dimana  $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \delta$ , akan dipilih  $\mathbf{x}_i$  sebagai solusi akhir jika memenuhi  $F(\mathbf{x}_i) > F(\mathbf{x}_j)$ , jika tidak, pilih  $\mathbf{x}_j$  sebagai solusi akhir dari fungsi objektif multimodal.
- Output:** Semua solusi yang memenuhi langkah 8 adalah solusi optimum dari permasalahan (4) yang merupakan representasi dari fungsi multimodal.

Pada beberapa kasus, terkadang kita hanya membutuhkan solusi optimum global meskipun dalam fungsi objektif multimodal terdapat sejumlah solusi optimum lokal. Untuk melokalisir solusi optimum global pada fungsi objektif, maka metode FPAC perlu dimodifikasi, yaitu:

- Tambahkan  $\rho (0 < \rho < 1)$  sebagai parameter *cut-off*
- Di langkah 3, tambahkan kriteria  $F(x_i) > \rho F(x^*)$  sebelum mengeksekusi fungsi *cluster*
- Di langkah 7, tambahkan kondisi  $F(x) > (1 - \rho)F(x^*)$  sebagai syarat seleksi.

### 3. HASIL DAN PEMBAHASAN

Untuk mengetahui keberhasilan metode yang diusulkan, maka metode *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC) perlu divalidasi. Validasi dilakukan dengan menggunakan *benchmark* fungsi multimodal yang dikompetisikan pada ajang CEC [10]. Pada setiap *benchmark*, algoritma dijalankan dalam sekali *running* dengan target menemukan semua solusi yang tersedia. Dalam penelitian ini, penulis mengkonstruksi program menggunakan *Octave-5.1.0* dan seluruh simulasi numeriknya dijalankan di komputer desktop dengan sistem operasi windows *10 home*, dengan spesifikasi: *processor intel core™ i5*, 8GB ram, dan 3,6 GHz CPU.

#### 3.1. Fungsi Tes 1, Second Minima

Fungsi Second minima untuk 2 dimensi adalah sebagai berikut:

$$g(x, y) = \frac{1}{2}(x^4 - 16x^2 + 5x) + \frac{1}{2}(y^4 - 16y^2 + 5y) \tag{5}$$

dimana  $D = \{(x, y)^T \in \mathbb{R}^2 \mid -4 \leq x \leq 4, -4 \leq y \leq 4\}$ .

Dengan menggunakan parameter  $m_{cl} = 300, k_{cl} = 10, m = 25, p = 0.5, \lambda = 1.5, \gamma = 0.1, \varepsilon = 10^{-5}, \delta = 10^{-2}$ , dan  $k_{maks} = 300$ , diperoleh hasil optimasi permasalahan (5) yang ditunjukkan pada tabel 1. Berdasarkan tabel 1, metode FPAC berhasil menemukan 5 solusi Optimum dari fungsi *Second minima* yang terdiri dari 4 solusi minimum 1 solusi maksimum.

Tabel 1 Solusi optimum fungsi *Second minima*

No	x	y	g(x,y)	No	x	y	g(x,y)
Minimum				Maksimum			
1	2.7468	-2.9035	-64.1956	1	0.1567	0.1567	0.3912
2	-2.9035	2.7468	-64.1956				
3	-2.9036	-2.9036	-78.3323				
4	2.7468	2.7468	-50.0589				

#### 3.2. Fungsi Tes 2, Six Hump Camel Back

Fungsi *Six hump camel back* untuk 2 dimensi adalah sebagai berikut:

$$g(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (4y^2 - 4)y^2 \tag{6}$$

dimana  $D = \{(x, y)^T \in \mathbb{R}^2 \mid -1.9 \leq x \leq 1.9, -1.1 \leq y \leq 1.1\}$ .

Dengan menggunakan parameter  $m_{cl} = 300, k_{cl} = 10, m = 25, p = 0.5, \lambda = 1.5, \gamma = 0.1, \varepsilon = 10^{-5}, \delta = 10^{-2}$ , dan  $k_{maks} = 300$ , diperoleh hasil optimasi permasalahan (6) yang ditunjukkan pada Tabel 2. Metode FPAC berhasil melokalisir 8 solusi optimum fungsi *Six hump camel back* yang terdiri dari 6 solusi minimum dan 2 solusi maksimum. Pada solusi minimum terdapat 4 minimum lokal dan 2 minimum global. Hasil yang ditemukan oleh FPAC lebih baik dari [10] dimana permasalahan (6) diinformasikan hanya memiliki 2 optimum lokal dan 2 optimum global.

Tabel 2 Solusi optimum fungsi Six hump camel back

No	x	y	g(x,y)	No	x	y	g(x,y)
Minimum				Maksimum			
1	-1.7036	0.7961	-0.2155	1	-1.2302	-0.1623	2.4963
2	-1.6071	-0.5687	2.1043	2	1.2302	0.1623	2.4963
3	0.0898	-0.7127	-1.0316				
4	1.6071	0.5687	2.1043				
5	1.7036	-0.7961	-0.2155				
6	-0.0898	0.7127	-1.0316				

3.3. Fungsi Tes 3, Rastrigin

Fungsi Rastrigin untuk n dimensi dengan domain yang dipilih sebagai berikut:

$$g(x) = 10d + \sum_{i=1}^d x_i^2 - 10 \cos(2\pi x_i) \tag{7}$$

dimana  $D = \{x = (x_1, \dots, x_n)^T \in \mathbb{R}^n \mid -1 \leq x_i \leq 1, i = 1, \dots, n\}$ .

Dengan menggunakan parameter  $m_{cl} = 300$ ,  $k_{cl} = 10$ ,  $m = 25$ ,  $p = 0.5$ ,  $\lambda = 1.5$ ,  $\gamma = 0.1$ ,  $\varepsilon = 10^{-5}$ ,  $\delta = 10^{-2}$ , dan  $k_{maks} = 300$ , diperoleh hasil optimasi permasalahan (7) untuk 2 dan 4 dimensi yang ditunjukkan pada tabel 3 dan 4. Berdasarkan tabel 3, metode FPAC berhasil menemukan semua solusi fungsi Rastrigin 2 dimensi yang terdiri dari 9 solusi minimum dan 4 solusi maksimum. Metode yang diusulkan juga divalidasi terhadap permasalahan multimodal untuk dimensi tinggi yaitu 4 dimensi. Berdasarkan tabel 4, metode FPAC berhasil melokalisasi 16 solusi maksimum dari fungsi rastrigin 4 dimensi.

Tabel 3 Solusi optimum fungsi Rastrigin 2 dimensi

No	X	y	g(x,y)	No	x	y	g(x,y)
Minimum				Maksimum			
1	0.9950	-0.9950	1.9899	1	-0.5025	-0.5025	40.5025
2	-0.9950	0.0000	0.9950	2	0.5025	0.5025	40.5025
3	0.0000	0.9950	0.9950	3	0.5025	0.5025	40.5025
4	0.0000	-0.9950	0.9950	4	-0.5025	0.5025	40.5025
5	-0.9950	-0.9950	1.9899				
6	0.9950	0.0000	0.9950				
7	0.9950	0.9950	1.9899				
8	0.0000	0.0000	0.0000				
9	-0.9950	0.9950	1.9899				

Tabel 4 Solusi maksimum fungsi Rastrigin 4 dimensi

No	x1	x2	x3	x4	g(x1,x2,x3,x4)
1	-0.5025	0.5025	0.5025	0.5025	81.0051
2	0.5025	-0.5025	-0.5025	0.5025	81.0051
3	-0.5025	-0.5025	0.5025	-0.5025	81.0051
4	0.5025	0.5025	-0.5025	0.5025	81.0051
5	0.5025	0.5025	0.5025	0.5025	81.0051
6	-0.5025	0.5025	-0.5025	0.5025	81.0051
7	0.5025	0.5025	0.5025	-0.5025	81.0051
8	-0.5025	0.5025	-0.5025	-0.5025	81.0051
9	0.5025	-0.5025	-0.5025	-0.5025	81.0051

10	0.5025	-0.5025	0.5025	0.5025	81.0051
11	-0.5025	-0.5025	-0.5025	0.5025	81.0051
12	-0.5025	0.5025	0.5025	-0.5025	81.0051
13	0.5025	-0.5025	0.5025	-0.5025	81.0051
14	-0.5025	0.5025	0.5025	0.5025	81.0051
15	0.5025	-0.5025	-0.5025	-0.5025	81.0051
16	-0.5025	-0.5025	-0.5025	-0.5025	81.0051

3.4. Fungsi Tes 4, Vincent

Fungsi Vincent untuk 2 dimensi adalah sebagai berikut:

$$g(x, y) = \frac{1}{2} (\sin(10 \log(x)) + \sin(10 \log(y))) \tag{8}$$

dimana  $D = \{(x, y)^T \in \mathbb{R}^2 \mid -0.25 \leq x \leq 10, -0.25 \leq y \leq 10\}$

Dengan menggunakan parameter  $m_{cl} = 1000$ ,  $k_{cl} = 10$ ,  $m = 25$ ,  $p = 0.5$ ,  $\lambda = 1.5$ ,  $\gamma = 0.1$ ,  $\varepsilon = 10^{-5}$ ,  $\delta = 10^{-2}$ , dan  $k_{maks} = 300$ , diperoleh hasil optimasi permasalahan (8) yang ditunjukkan pada tabel 5. Metode FPAC berhasil melokalisir semua solusi optimum fungsi Vincent 2 dimensi yaitu 36 solusi optimum. Hal ini sesuai dengan informasi dalam [10] bahwa fungsi Vincent memiliki  $6^D$  optimum global dan 0 optimum lokal.

Tabel 5 Solusi maksimum fungsi Vincent 2 dimensi

No	x	y									
1	4.1112	0.6242	10	2.1933	4.1112	19	1.1701	2.1933	28	0.6242	0.3330
2	0.6242	2.1933	11	4.1112	0.3330	20	7.7063	0.3330	29	0.3330	0.6242
3	4.1112	1.1701	12	0.3330	1.1701	21	1.1701	7.7063	30	7.7063	7.7063
4	7.7063	1.1701	13	2.1933	0.6242	22	0.3330	4.1112	31	0.3330	0.3330
5	0.6242	4.1112	14	0.6242	7.7063	23	7.7063	0.6242	32	7.7063	4.1112
6	0.3330	2.1933	15	1.1701	1.1701	24	0.6242	0.6242	33	0.6242	1.1701
7	4.1112	2.1933	16	4.1112	7.7063	25	2.1933	2.1933	34	1.1701	0.6242
8	0.3330	7.7063	17	1.1701	0.3330	26	2.1933	1.1701	35	7.7063	2.1933
9	4.1112	4.1112	18	2.1933	0.3330	27	1.1701	4.1112	36	2.1933	7.7063

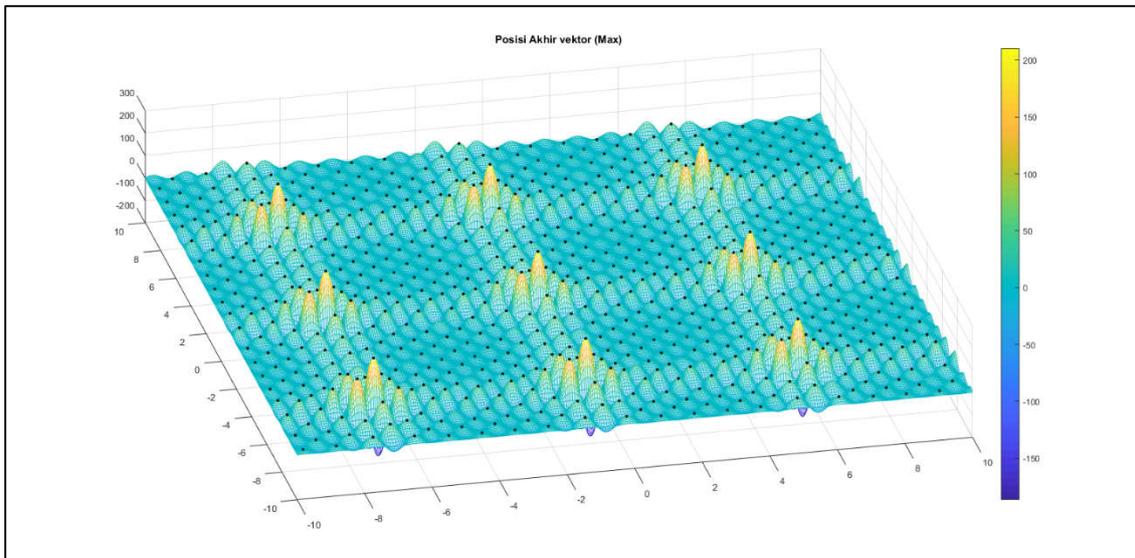
3.5. Fungsi tes 5, Shubert

Fungsi Shubert untuk n dimensi adalah sebagai berikut:

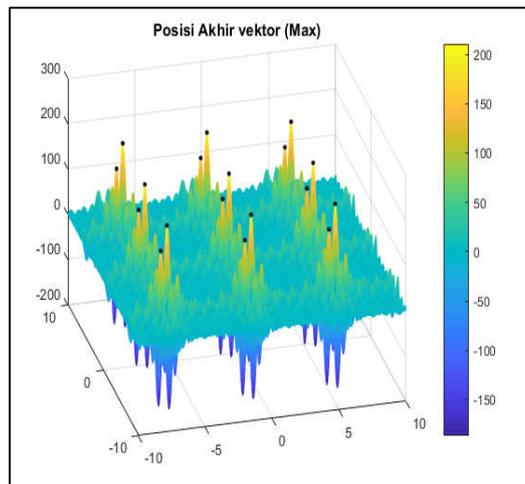
$$g(x, y) = - \prod_{i=1}^n \sum_{j=1}^5 j \cos[(j+1)x_i + j] \tag{9}$$

dimana  $D = \{x = (x_1, \dots, x_n)^T \in \mathbb{R}^n \mid -10 \leq x_i \leq 10, i = 1, \dots, n\}$ .

Dengan menggunakan parameter  $m_{cl} = 1000$ ,  $k_{cl} = 20$ ,  $m = 50$ ,  $p = 0.5$ ,  $\lambda = 1.5$ ,  $\gamma = 0.1$ ,  $\varepsilon = 10^{-5}$ ,  $\delta = 10^{-2}$ ,  $\rho = 0.3$  dan  $k_{maks} = 500$ , diperoleh hasil optimasi permasalahan (9) yang ditunjukkan pada tabel 6. Berdasarkan informasi dalam [10] bahwa fungsi Shubert memiliki  $D \cdot 3^D$  optimum global dan banyak optimum lokal. Metode FPAC berhasil menunjukkan bahwa fungsi Shubert memiliki banyak optimum lokal yang ditunjukkan oleh gambar 2. Untuk melokalisir solusi optimum global dari sejumlah optimum lokal yang jumlahnya belum terkuantisasi maka parameter cut-off digunakan. Dengan menggunakan parameter cut-off, metode FPAC sukses menemukan 18 optimum global yang ditunjukkan oleh gambar 3.



Gambar 2 Optimasi fungsi *Shubert* 2 dimensi menggunakan metode *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC)



Gambar 3 Optimasi fungsi *Shubert* 2 dimensi menggunakan metode *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC) dengan tambahan parameter *cut-off*

Tabel 6. Solusi maksimum global untuk fungsi *Shubert* 2 dimensi dengan tambahan paramter *cut-off*.

No	x	y	g(x,y)	No	x	y	g(x,y)	No	x	y	g(x,y)
1	-7.7083	-7.7083	165.660	7	-1.4251	-7.7083	165.660	13	4.8581	-7.7083	165.660
2	-7.0835	-7.0835	210.482	8	-0.8003	-7.0835	210.482	14	5.4829	-7.0835	210.482
3	-7.7083	-1.4251	165.660	9	-1.4251	-1.4251	165.660	15	4.8581	-1.4251	165.660
4	-7.0835	-0.8003	210.482	10	-0.8003	-0.8003	210.482	16	5.4829	-0.8003	210.482
5	-7.7083	4.8581	165.660	11	-1.4251	4.8581	165.660	17	4.8581	4.8581	165.660
6	-7.0835	5.4829	210.482	12	-0.8003	5.4829	210.482	18	5.4829	5.4829	210.482

#### 4. KESIMPULAN DAN SARAN

Metode yang diusulkan yaitu *Flower Pollination Algorithm* dengan teknik *Clustering* (FPAC) berhasil menemukan semua solusi dari masing-masing fungsi multimodal yang diwakili oleh 5 fungsi *benchmark* yaitu *Second minima*, *Six hump camel back*, *Rastrigin*, *Vincent* dan *Shubert* dalam sekali *running* baik untuk permasalahan dimensi rendah (2 dimensi) maupun dimensi tinggi (4 dimensi). Pada fungsi *Vincent* dan *Shubert* 2 dimensi, metode FPAC menunjukkan hasil yang sama dengan [10]. Akan tetapi pada fungsi *Six hump camel back*, metode yang diusulkan memberikan hasil yang lebih baik dari [10]. Meskipun metode FPAC dapat melokalisasi semua solusi dari masing-masing fungsi multimodal yang diujikan, tetapi metode ini perlu dikembangkan agar dapat menyelesaikan permasalahan *Multiobjective multimodal optimization* dalam [1] dan fungsi multimodal komposit dalam [10].

#### DAFTAR PUSTAKA

- [1] C. Yue, B. Qu, K. Yu, J. Liang, and X. Li, "A novel scalable test problem suite for multimodal multiobjective optimization," *Swarm Evol. Comput.*, vol. 48, no. March, pp. 62–71, 2019.
- [2] X. Lin, W. Luo, and P. Xu, "Differential evolution for multimodal optimization with species by nearest-better clustering," *IEEE Trans. Cybern.*, 2019.
- [3] S. I. A. Idrus, H. Syahputra, and M. Firdaus, "Modification of species-based differential evolution for multimodal optimization," *AIP Conf. Proc.*, vol. 1691, no. 1, p. 030012, 2015.
- [4] X. Yang, *Nature-Inspired Optimization Algorithms*, 1st ed. Elsevier, 2014.
- [5] N. Nekouie and M. Yaghoobi, "A new method in multimodal optimization based on firefly algorithm," *Artif. Intell. Rev.*, vol. 46, no. 2, pp. 267–287, 2016.
- [6] J. Gálvez, E. Cuevas, and O. Avalos, "Flower pollination algorithm for multimodal optimization," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 627–646, 2017.
- [7] N. Kushwaha and M. Pant, "Modified particle swarm optimization for multimodal functions and its application," *Multimed. Tools Appl.*, vol. 78, no. 17, pp. 23917–23947, 2019.
- [8] K. A. Sidarto, A. Kania, and N. Sumarti, "Finding multiple solutions of multimodal optimization using spiral optimization algorithm with clustering," *Mendel*, vol. 23, no. 1, pp. 95–102, 2017.
- [9] X. Yang, "Flower Pollination Algorithm for Global Optimization," in *International conference on unconventional computing and natural computation*, Berlin, Heidelberg: Springer, 2012, pp. 240–249.
- [10] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark Functions for CEC ' 2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization," *Tech. Report, Evol. Comput. Mach. Learn. Group, RMIT Univ. Aust.*, pp. 1–10, 2016.
- [11] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes," *Phys. Rev. E*, vol. 49, no. 5, p. 4677, 1994.
- [12] R. Hadi, "Optimasi Fitur dalam Klasterisasi Mahasiswa Program Studi Sistem Informasi Dengan Algoritma Genetik," *Techno.COM*, vol. 16, no. 3, pp. 249–255, 2017.