

# SISTEM PENJADWALAN ANGGOTA PADA APLIKASI EVENT MANAGEMENT MENGGUNAKAN ALGORITMA PARTICLE SWARM OPTIMIZATION BERBASIS WEB

## *MEMBER SCHEDULING SYSTEM USING PARTICLE SWARM OPTIMIZATION ALGORITHM IN WEB-BASED EVENT MANAGEMENT APPLICATION*

Muhammad Izzah Aeman<sup>1</sup>, Roswan Latuconsina<sup>2</sup>, Casi Setianingsih<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

<sup>1</sup>mizzahae@student.telkomuniversity.ac.id, <sup>2</sup>roswan@telkomuniversity.ac.id,

<sup>3</sup>setiacasie@telkomuniversity.ac.id

---

### Abstrak

Dalam sebuah *event management*, salah satu masalah yang sering ditemukan adalah penjadwalan. Penjadwalan yang dibuat secara manual di sebuah kepanitiaan khususnya mahasiswa, sering menjadi tidak optimal, dan terjadi ke tidak cocokkan antara jadwal kesibukan mahasiswa dengan jadwal acara.

Untuk mengatasi masalah tersebut penelitian ini merancang sebuah model penjadwalan anggota panitia secara otomatis berbasis algoritma Particle Swarm Optimization (PSO). Model penjadwalan otomatis ini bertujuan menghasilkan penjadwalan anggota yang optimal dalam sebuah *event management*. Kandidat partikel merepresentasikan jadwal *event*, sedangkan dimensi merepresentasikan divisi anggota, dan posisi partikel merepresentasikan anggota. Populasi partikel dibangkitkan di awal iterasi dengan nilai acak dan pada setiap iterasi, partikel-partikel tersebut memperbaiki posisinya menuju posisi terbaik. Proses ini dilakukan untuk setiap penjadwalan anggota yang akan dijadwalkan, sehingga diperoleh sebuah jadwal yang optimal.

Berdasarkan hasil pengujian, dilakukan percobaan perubahan *inertia*, pembelajaran kognitif, dan sosial untuk digunakan dalam PSO ini. Pada penelitian ini, nilai *inertia* yang optimal 0,9 sedangkan nilai pembelajaran kognitif, dan sosial yaitu 1,4. Dalam Percobaannya Nilai rata-rata iterasi saat berhenti 118,6. Hasil yang didapatkan pada setiap percobaan akan berbeda-beda dikarenakan nilai acak yang digunakan pada setiap percobaan berbeda. Pada penelitian ini algoritma PSO berhasil menghasilkan penjadwalan anggota panitia yang sesuai dengan penjadwalan anggota panitia

**Kata kunci :** *Event Management, Penjadwalan, Mahasiswa, Particle Swarm Optimization.*

---

### Abstract

In event management, one of the problems that are often found is scheduling. Scheduling made manually in a committee, especially for students, often becomes not optimal and does not match the student's busy schedule with the event schedule.

This study designed an automatic scheduling model for committee members based on Particle Swarm Optimization (PSO) to overcome this problem. This automatic scheduling model aims to produce optimal member scheduling in event management. Candidate particles represent the schedule of events, while dimensions represent member divisions, and particle positions represent members. The population of particles is generated at the beginning of the iteration with random values, and at each iteration, the particles improve their position towards the best position. This process is carried out for each scheduling member who will be invited to obtain the optimal schedule.

Based on the test, an experiment of inertial change, cognitive learning, and social change was conducted to be used in this PSO. In this study, the optimal value of inertia is 0.9, while the value of cognitive and social learning is 1.4. In the experiment, the average iteration value at stopping was 118.6. The results obtained in each trial will be different because the random values used in each experiment are different. In this study, the PSO algorithm succeeded in producing the scheduling of committee members according to the schedule of committee members.

**Keywords: Event Management, Scheduling, College Student, Particle Swarm Optimization**

---

## 1. Pendahuluan

*Event* adalah sebuah program atau proyek yang akan dilakukan secara terencana untuk suatu tujuan [1]. Acara dapat diselenggarakan dengan adanya sebuah kepastian jadwal yang dapat mengatur acara. *Scheduling* atau penjadwalan adalah salah satu masalah utama bagi sebuah kepanitiaan atau *event management* [2]. Penjadwalan yang baik merupakan salah satu hal yang dapat menghasilkan kelancaran pada sebuah acara.

Berdasarkan hasil survei yang telah dilakukan kepada mahasiswa Universitas Telkom, terdapat 86,2% responden yang membutuhkan sebuah fitur penjadwalan anggota. Responden juga memiliki beberapa kendala yang dihadapi dalam pencatatan atau pengelolaan penjadwalan, yaitu:

1. Terjadi ke tidak cocokkan antara jadwal kuliah mahasiswa dengan jadwal acara.
2. Penyampaian informasi penjadwalan kepada anggota kepanitiaan yang tidak efektif dan tidak lengkap.

Dengan adanya masalah penjadwalan di atas, dibutuhkan suatu teknologi informasi yang dapat membantu kepanitiaan dalam mengatur penjadwalannya. Cara mengatasi permasalahan di atas adalah dengan membuat sebuah aplikasi sistem *Event Management*, khususnya di bagian penjadwalan anggota panitia dengan menggunakan Algoritma *Particle Swarm Optimization*. Algoritma tersebut digunakan untuk menghasilkan jadwal dengan menggunakan nilai *fitness* yang paling optimal [3]. Penelitian ini menggunakan Algoritma *Particle Swarm Optimization* untuk penjadwalan anggota panitia, dikarenakan menurut penelitian sebelumnya PSO dapat menyelesaikan masalah tentang penjadwalan [4].

## 2. Dasar Teori /Material dan Metodologi/perancangan

### 2.1 Penjadwalan

Penjadwalan adalah proses pengambilan keputusan yang digunakan secara teratur di banyak industri manufaktur dan jasa. Ini berkaitan dengan alokasi sumber daya untuk tugas-tugas selama periode waktu tertentu dan tujuannya adalah untuk mengoptimalkan satu atau lebih tujuan. Penjadwalan, sebagai proses pengambilan keputusan, memainkan peran penting dalam sebagian besar sistem produksi di sebagian besar lingkungan pemrosesan informasi [5]. Ini juga penting bagi sebuah kepanitiaan atau event management [2].

Penyusunan jadwal anggota dalam sebuah kegiatan kepanitiaan atau event management, tentu tidak dapat dilakukan sembarangan, karena keberhasilan acara yang efektif ditentukan oleh proses penjadwalan anggota yang optimal. Dalam penyusunan jadwal anggota, hal yang harus dilakukan adalah pendataan anggota panitia, sanggup mengikuti jadwal acara atau tidak. Ini dilakukan agar nantinya tidak terjadi bentrok antar kegiatan kepanitiaan dengan kesibukan panitia itu sendiri. Sering kali proses ini memakan waktu yang tidak sebentar dikarenakan prosesnya yang manual. Penyampaian informasi jadwal acara secara lengkap dan optimal, juga penting agar tidak terjadinya miss komunikasi antar kepanitiaan.

### 2.2 Particle Swarm Optimization (PSO)

Optimasi merupakan aktivitas untuk mendapatkan hasil yang terbaik atau optimal (nilai efektif yang dapat dicapai) dari pilihan yang tersedia. Tujuan dari setiap keputusan adalah untuk meminimalkan usaha yang dilakukan atau memaksimalkan keuntungan yang diperoleh [6]. Tujuan dari Optimasi Penjadwalan anggota panitia adalah untuk merencanakan anggota acara ke dalam penjadwalan acara. Dengan merencanakan penjadwalan anggota panitia dengan baik acara maka acara akan berjalan dengan lancar. Dengan menggunakan Algoritma *Particle Swarm Optimization* panitia acara dapat terbantu dalam pembuatan penjadwalan anggota panitia.

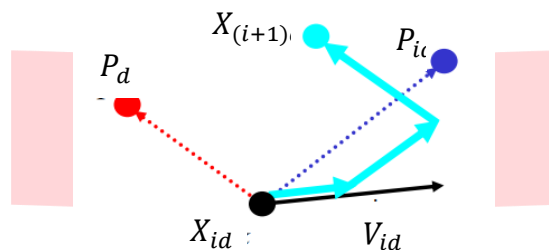
Algoritma *Particle Swarm Optimization* (PSO) diperkenalkan oleh Dr. R. C. Eberhart dan Dr. J. Kennedy pada tahun 1995, merupakan algoritma optimasi yang terinspirasi oleh perilaku sosial hewan, seperti kawanan burung atau kawanan ikan [7]. Sejak diperkenalkan pertama kali, algoritma PSO berkembang cukup pesat, baik dari sisi aplikasi maupun dari sisi pengembangan metode yang digunakan pada algoritma tersebut (R.L., Haupt, S.E Haupt, 2004) [8]. Algoritma PSO menstimulasikan perilaku sosial seekor burung. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Kata partikel menunjukkan, misalnya, seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan

yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut meskipun lokasi mereka jauh di kelompok tersebut[9].

PSO dimulai dengan sekumpulan partikel (solusi) yang dibangkitkan secara acak. Setiap partikel kemudian dievaluasi kualitasnya menggunakan fungsi *fitness*. Selanjutnya, partikel-partikel akan terbang mengikuti partikel yang optimum. Pada saat iterasi, setiap partikel diperbarui mengikuti dua nilai terbaik. Yang pertama adalah *fitness* terbaik yang dicapai oleh satu partikel. Nilai *fitness* terbaik ini dilambangkan dengan  $P_{best}$  dan disimpan di memori. Sedangkan nilai terbaik yang kedua adalah *fitness* terbaik yang dicapai oleh semua partikel dalam tipologi ke tetangga. Indeks  $G_{best}$  digunakan untuk menunjukkan partikel dengan *fitness* terbaik. Berikut ini merupakan model matematika yang menggambarkan mekanisme memperbarui kecepatan partikel dan posisi partikel [10].

$$V_{id} = \omega V_{id} + C_1 r_1 (P_{id} - X_{id}) + C_2 r_2 (P_d - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$



Gambar 1. Ilustrasi dari Algoritma PSO [11]

Berikut penjelasannya:

1.  $i$  = indek partikel ke- $i$
2.  $d$  = indek dimensi ke- $d$
3.  $\omega$  = bobot inersia
4.  $C_1$  = Parameter kognitif
5.  $C_2$  = Parameter sosial
6.  $r_1$  dan  $r_2$  = angka acak dengan range  $[0,1]$
7.  $P_{id}$  = Pbest partikel ke-  $i$  dimensi ke- $d$
8.  $P_d$  = Gbest dimensi ke- $d$
9.  $V_{id}$  = Kecepatan partikel ke- $i$  dimensi ke- $d$
10.  $X_{id}$  = Posisi partikel ke- $i$  dimensi ke- $d$

Tidak seperti Genetic Algorithm (GA), PSO tidak memiliki operator evolusi, seperti crossover dan mutasi. Baris dalam matriks disebut partikel (sama dengan kromosom GA). Setiap partikel bergerak sekitar di permukaan partikel dengan kecepatan [12].

Berikut istilah yang digunakan dalam penerpaan algoritma PSO [10]:

1. Swarm = populasi dari sekawasan partikel.
2. Particle = individu pada suatu swarm. Setiap partikel mempresentasikan suatu solusi dari permasalahan yang diselesaikan.
3. Pbest = suatu partikel yang menunjukkan posisi terbaik (Partikel Best).
4. Gbest = posisi terbaik dari seluruh partikel yang ada dalam suatu swarm (Global Best).
5. Velocity = kecepatan yang dimiliki oleh setiap partikel dalam menentukan arah perpindahan suatu partikel untuk memperbaiki posisi semula.
6.  $C_1$  = merupakan konstanta pembelajaran kognitif.
7.  $C_2$  = konstanta pembelajaran sosial.

Pada penelitian PSO ini digunakan untuk mencari solusi optimal untuk penjadwalan anggota. Kecerdasan Buatan yang harus memenuhi batasan-batasan yang telah ditentukan.

### 2.2.1. Parameter Algoritma PSO

Untuk parameter yang digunakan dalam proses algoritma PSO adalah sebagai berikut [13]:

1. Jumlah partikel (*Number of particles*) merupakan faktor yang dianggap sangat penting dalam melakukan penyelesaian masalah.
2. Bobot Inersia (*Inertia Weight*), memainkan peran yang sangat penting dalam kecepatan partikel dari algoritma PSO.
3. Faktor pembelajaran (*Learning factors*). Parameter  $c_1$  merupakan pengakuan koefisien, sedangkan  $c_2$  adalah komponen sosial.

4. Rentang dan dimensi partikel (*Range and dimension of particles*). Dimensi partikel dan rentang ditentukan berdasarkan masalah yang dioptimalkan.
5. Kecepatan (*Velocity*) merupakan perubahan maksimum pada setiap partikel, dapat diambil selama iterasi yang didefinisikan sebagai kecepatan maksimum.
6. Menghentikan kondisi (*stopping condition*) merupakan salah satu cara apabila kriteria yang dicari sudah tercapai.

### 2.2.2. Proses Algoritma PSO

Berikut Proses dari Algoritma PSO [10]:

1. Bangkitkan posisi awal sejumlah partikel sekaligus kecepatan awalnya secara acak.
2. Evaluasi nilai *fitness* dari masing-masing partikel berdasarkan posisinya.
3. Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai *Gbest*. Untuk setiap partikel, *Pbest* awal akan sama dengan posisi awal.

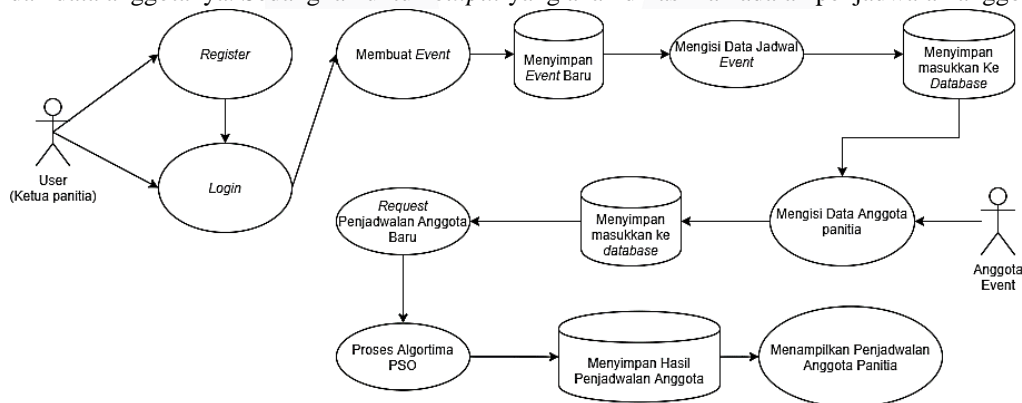
Ulangi langkah berikut sampai *stopping criteria* dipenuhi

- A. Menggunakan *Pbest* dan *Gbest* yang ada, perbarui kecepatan setiap partikel menggunakan persamaan (2,1). Lalu dengan kecepatan baru yang didapat, perbarui posisi setiap partikel menggunakan persamaan (2,2).
- B. Evaluasi *fitness* dari setiap partikel
- C. Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai *Best*. Untuk setiap partikel, tentukan *Pbest* dengan membandingkan posisi sekarang dengan *Pbest* dari iterasi sebelumnya.
- D. Cek *stopping criteria*. Jika dipenuhi berhenti. Jika tidak kembali ke A.

## 3. Metodologi Penelitian

### 3.1. Gambaran Umum Sistem

Gambaran umum sistem pada penelitian ini merupakan gambaran mengenai sistem pada Aplikasi *Event Management* dalam penjadwalan anggota menggunakan algoritma *Particle Swarm Optimization* (PSO). Adapun *input* yang diberikan oleh *user* (ketua panitia) adalah data jadwal *event* dan data anggotanya. Sedangkan untuk *output* yang akan dihasilkan adalah penjadwalan anggota.



Gambar 2. Gambaran Umum Sistem

- a. Seluruh user dapat memulai aplikasi dengan mendaftar, setelah itu *login* ke aplikasi.
- b. User membuat *event* baru, dengan mengisi formulir *event* yang berisi nama, penyelenggara, tempat, *budget*, dan kode tambah anggota *event*.
- c. Kemudian *user* dapat mengisi data jadwal *event* yang berisi nama jadwal *event*, awal dan akhir waktu acaranya.
- d. Selanjutnya *user* mengisi data anggota panitia yang berisi nama anggota, nomor telepon anggota, divisi anggota dan kesanggupan mengikuti suatu jadwal *event* (Anggota *Event* juga dapat mengisi datanya sendiri dengan kode tambah anggota *event*).
- e. Setelah *user* mengisi data jadwal *event* dan data anggota *event*, *user* dapat *generate* penjadwalan anggota panitia baru, setelah itu sistem membaca data jadwal anggotanya dan menghasilkan penjadwalan anggota panitia ke database dengan menggunakan algoritma PSO.
- f. Aplikasi akan menghasilkan penjadwalan anggota panitia dari hasil algoritma PSO.

### 3.2. Aturan Penjadwalan Anggota

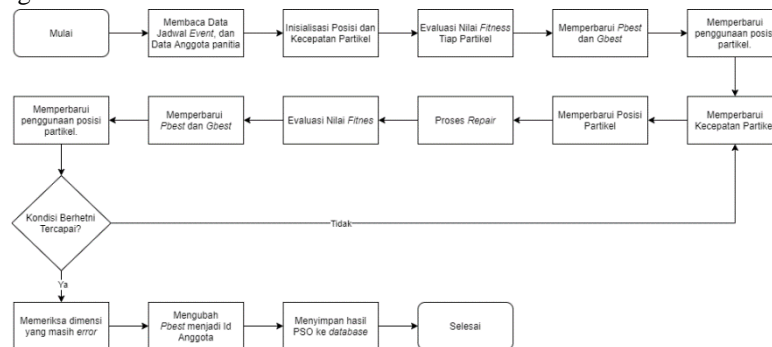
Aturan diperlukan agar mendapatkan penjadwalan anggota yang optimal. Berikut adalah aturan saat membuat penjadwalan anggota di proses Algoritma PSO.

Tabel 1. Aturan Membuat Penjadwalan Anggota

No.	Aturan
1	Anggota tidak boleh mendapatkan jadwal yang bertabrakan dengan jadwal acara
2	Anggota tidak boleh mendapatkan jadwal melebihi dari batas maksimal
3	Seluruh panitia mendapatkan jadwal

### 3.3. Pemodelan PSO

Pada penelitian ini, PSO digunakan untuk mencari solusi optimal penjadwalan kepanitiaan kecerdasan buatan yang harus memenuhi batasan-batasan yang telah ditentukan di tabel 1 Berikut proses dari Algoritma PSO:



Gambar 3. Flow Chart Proses PSO

#### 3.3.1. Membaca Data Jadwal Event dan Data Anggota Event

Sebelum memulai algoritma PSO, sistem tersebut membaca data jadwal *event* dan data anggota *event*. Data jadwal *event* yang dibaca yaitu jumlah jadwal *event* yang akan dijadikan jumlah populasi partikel PSO. Sedangkan data anggota panitia di baca seluruhnya. Setelah membaca data anggota panitia, sistem akan menghitung jumlah anggota per-divisi dan memperbarui pointer di data anggota panitia. Pointer ini berguna mengurutkan data anggota panitia berdasarkan di divisi.

Selanjutnya menghitung batas maksimal anggota panitia (posisi partikel) mendapatkan jadwal. Untuk rumusnya sebagai berikut:

$$f = \left( \frac{\text{jumlah anggota per - divisi}}{\text{jumlah jadwal event}} \right) + 1 \quad (3)$$

#### 3.3.2. Inisialisasi

Dalam PSO ini partikel merepresentasikan jadwal *event*, sedangkan dimensi merepresentasikan divisi anggota, dan posisi partikel merepresentasikan anggota. Jadi misal partikel ke 3 dimensi ke 6 posisinya = 3, maka jadwal *event* ke 3 memiliki anggota berdivisi 6 yang memiliki pointer 3.

Pada Proses inisialisasi ini dilakukan terhadap jumlah populasi partikel, dimensi, iterasi, *inertia*( $\omega$ ), Parameter kognitif ( $C_1$ ), Parameter sosial ( $C_2$ ), posisi awal partikel ( $X_{(t)}$ ), jumlah iterasi, dan kecepatan awal partikel ( $V_{(t)}$ ). Berikut inisialisasi awal dalam PSO ini:

1. Populasi partikel ditentukan dari banyaknya jumlah *event*. Dalam kasus ini yaitu 20 populasi partikel.
2. Dimensi ditentukan dari banyaknya jumlah divisi yaitu 7.
3. Iterasi yang digunakan yaitu 500.
4. *inertia*( $\omega$ ) yaitu 0,9. Sedangkan Parameter kognitif ( $C_1$ ) dan parameter sosial ( $C_2$ ) yaitu 1,4.
5. Posisi awal partikel ( $X_{(t)}$ ) ditentukan dengan acak, dan memiliki batasan-batasan yang berbeda, mulai dari 1 hingga batas maksimalnya tergantung dari jumlah anggota per-divisi. Misal posisi awal partikel dimensi ke 3, maka posisi awal partikel tidak boleh kurang dari 1 dan tidak boleh lebih dari jumlah anggota divisi ke 3.
6. kecepatan awal partikel ( $V_{(t)}$ ) ditentukan dengan acak juga dan memiliki batasan-batasan yang sama dengan posisi awal partikel.

#### 3.3.3. Evaluasi Nilai Fitness Tiap Partikel

Di penelitian ini, nilai *fitness* tiap partikel dihitung melalui pengecekan aturan yang telah ditentukan di tabel 1. Pencarian penjadwalan anggota panitia yang optimal memperhatikan aturan yang telah ditentukan

Berdasarkan aturan pada tabel 1 di kasus ini nilai *fitness* diambil dari banyaknya batasan yang dilanggar sehingga semakin kecil jumlah pelanggaran yang dilanggar maka solusi yang dihasilkan akan semakin baik. Agar tidak terjadi nilai *fitness* yang tak terhingga maka jumlah total semua pelanggaran akan ditambahkan 1.

$$f = \sum \text{Tidak Dapat Mengikuti Acara} + \sum \text{Melebihi Maksimal} \quad (4)$$

Maksud dari tidak dapat mengikuti acara adalah saat posisi partikel (anggota panitia) jika dia tidak dapat mengikuti acara maka nilai *fitness* ditambah, sedangkan dari melebihi maksimal adalah saat penggunaan posisi partikel (anggota panitia) melebihi batas maksimal dari yang digunakan maka nilai *fitness* ditambah.

### 3.3.4. Memperbarui *Pbest* dan *Gbest*

Perbarui *Pbest* dengan mencari nilai *fitness* minimum tiap partikel dan bandingkan dengan nilai *fitness* sebelumnya. Jika  $f(Pbest_i) > f(X_i)$ , maka  $f(Pbest_i) = X_i$ . Tetapi jika saat iterasi = 0, maka  $f(Pbest_i) = X_i$ .

Perbarui *Gbest* dengan mengevaluasi keseluruhan *fitness Pbest* dan mengambil yang paling minimum atau sama dengan 0.

### 3.3.5. Memperbarui Penggunaan Posisi Partikel (Penggunaan Anggota)

Setelah melakukan *update Pbest*, dan jika pertama kali nilai  $f(Pbest_i) = 0$ , akan dilakukan perbaruan penggunaan posisi partikel. Perbaruan penggunaan posisi partikel ini diperlukan agar penggunaan posisi partikel tidak melebihi batas. Lakukan hal tersebut ke seluruh partikel yang pertama kali mendapatkan nilai  $f(Pbest_i) = 0$ .

### 3.3.6. Memperbarui Kecepatan Partikel

Melakukan perbaruan kecepatan Partikel dengan rumus (1) ke, seluruh partikel.

### 3.3.7. Memperbarui Posisi Partikel

Melakukan perbaruan posisi partikel dengan rumus (2) ke seluruh partikel.

### 3.3.8. Proses *Repair*

Pada penelitian ini proses *repair* mencegah tidak terkontrolnya saat melakukan *update* posisi partikel dalam representasi partikel tersebut. proses *repair* pertama dilakukan dengan memeriksa apakah partikel tersebut kurang dari batas kurang atau melebihi batas maksimal dari posisi partikel. Dengan adanya proses *repair* ini representasi partikel yang berupa bilangan real kemudian dibulatkan menjadi integer dan merepresentasikan id dari data *pointer* anggota sehingga menjadi tidak liar.

Proses *repair* juga dilakukan dengan cara mengecek seluruh dimensi di partikel yang memiliki nilai *error* atau yang menyebabkan menambahnya nilai *fitness*. Untuk proses pengecekannya, hampir sama dengan evaluasi nilai *fitness* tiap partikel akan tetapi jika di evaluasi nilai *fitness* itu menambah nilai *fitness*, di proses *repair* akan melakukan proses *repair* posisi partikel di dimensi (divisi) partikel tersebut. Hal ini dilakukan agar representasi posisi partikel dapat menghasilkan solusi yang optimum.

### 3.3.9. Evaluasi Nilai *Fitness* Tiap Partikel

Lakukan hal yang sama dengan Evaluasi Nilai *Fitness* Tiap Partikel sebelumnya.

### 3.3.10. Memperbarui *Pbest* dan *Gbest*

Hampir sama yang dilakukan saat memperbarui *Pbest* sebelumnya, hanya saja jika nilai  $f(Pbest_i)$  belum sama dengan 0, maka dilakukan *update f(Pbest<sub>i</sub>)* secara manual. Cara memperbaruinya hampir sama dengan evaluasi nilai *fitness* tiap partikel akan tetapi jika di evaluasi nilai *fitness* itu menggunakan posisi partikel, di proses *update* ini akan menggunakan *Pbest*. Hal ini dilakukan karena penggunaan posisi partikel akan selalu *update*, jadi misalnya di literasi 4  $f(-Pbest_i)$  awalnya = 2, di literasi 10 bisa saja  $f(Pbest_i) = 3$ , dikarenakan di dimensi *Pbest* terdapat salah satu penggunaan posisi partikelnya melebihi batas maksimal. Sedangkan untuk Perbaruan *Gbest* sama dengan sebelumnya.

### 3.3.11. Memperbarui Penggunaan Posisi Partikel

Lakukan hal yang sama dengan memperbarui penggunaan posisi partikel sebelumnya.

### 3.3.12. Pemeriksaan Kondisi

Jika seluruh partikel  $f(Pbest_i) = 0$  atau jumlah iterasinya sudah tercapai, maka dapat melanjutkan tahap berikutnya. Sebaliknya jika kondisi tidak tercapai maka lakukan pengulangan perbaruan kecepatan partikel sebelumnya hingga memperbarui penggunaan posisi partikel.

### 3.3.13. Memeriksa Posisi Partikel Yang Masih *Error*

Jika jumlah iterasinya sudah tercapai dan masih terdapat partikel  $f(Pbest_i)$  tidak sama dengan 0 maka partikel tersebut akan diperiksa. Pemeriksaannya hampir sama dengan evaluasi nilai *fitness* tiap partikel akan tetapi jika di evaluasi nilai *fitness* itu menambah nilai *fitness*, di proses ini akan memperbarui *Pbest* di dimensi (divisi) partikel tersebut menjadi 0.

### 3.3.14. Mengubah *Pbest* Menjadi Id Anggota

Proses perubahan dari *Pbest* menjadi id anggota di lakukan agar dapat merepresentasikan anggota di partikel tersebut.

### 3.3.15. Menyimpan Hasil PSO ke *Database*

Setelah mengubah *Pbest* menjadi id anggota panitia barulah seluruh *Pbest* partikel di simpan ke *database*, yang menjadi penjadwalan anggota panitia.

## 4. Pengujian Algoritma PSO

Untuk dapat membuktikan bahwa hasil dari algoritma PSO memenuhi aturan pada tabel 1 dan berjalan dengan seharusnya. Maka dilakukan uji validasi dengan melakukan percobaan perubahan parameter. Berikut adalah pengujian saat mengubah nilai *inertia*,  $C_1$ , dan  $C_2$ .

Tabel 2. Parameter Pengujian Algoritma PSO

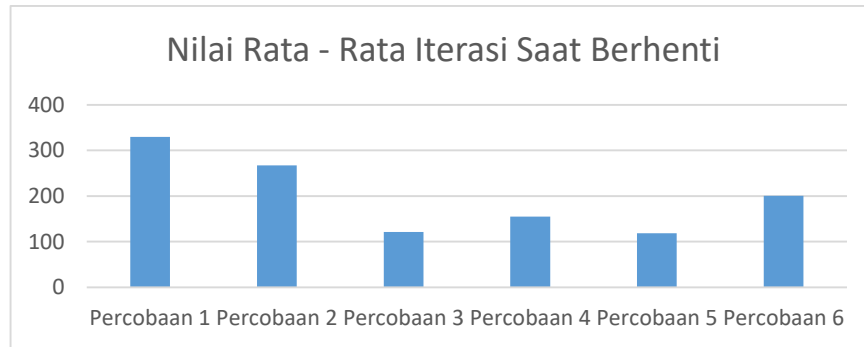
No	Parameter Pengujian	Percobaan n 1	Percobaan n 2	Percobaan n 3	Percobaan n 4	Percobaan n 5	Percobaan n 6
1	Jumlah Partikel	20	20	20	20	20	20
2	Dimensi	7	7	7	7	7	7
3	Iterasi	500	500	500	500	500	500
4	$\omega$ ( <i>inertia</i> )	0,7	0,8	0,9	0,9	0,9	0,9
5	$C_1$ (pembelajaran kognitif)	1,4	1,4	1,4	1,1	1,4	1,7
6	$C_2$ (pembelajaran sosial)	1,4	1,4	1,4	1,7	1,4	1,1

Pada percobaan tabel 2 akan dilakukan sebanyak 10 untuk mencari nilai *nilai inertia*,  $C_1$ , dan  $C_2$  mana yang paling optimal untuk kasus ini. Berikut adalah hasil dari percobaan di atas:

Tabel 3. Hasil Pengujian Algoritma PSO

No	Hasil	Percobaan n 1	Percobaan n 2	Percobaan n 3	Percobaan n 4	Percobaan n 5	Percobaan n 6
1	Mendapatkan Seluruh Partikel ber- <i>fitness</i> = 0	5	7	10	10	10	8
2	Nilai rata-rata iterasi saat berhenti	329,4	267,4	121,6	155,2	118,6	200,4

Gambar 4. Grafik Nilai Rata-Rata Iterasi Saat Berhenti Di Pengujian



Berdasarkan hasil dari tabel 3 dapat dilihat bahwa seluruh percobaan partikel pernah mendapatkan  $fitness = 0$ , itu, menandakan bahwa percobaan di tabel 3, dapat memenuhi aturan dari tabel 1 dan berhasil menghasilkan penjadwalan anggota yang baik. Hasil percobaan ini dapat berubah-ubah dikarenakan saat inisialisasi awal partikel menggunakan nilai acak. Dari hasil percobaan di atas, disimpulkan bahwa percobaan 5 yang tercepat. Di karena kan dari grafik di gambar 4 percobaan 5 paling sedikit di dibandingkan dengan yang lain, yang berarti saat melakukan percobaan 5 akan lebih cepat mendapatkan nilai optimal.

## 5. Kesimpulan Dan Saran

### 5.1. Kesimpulan

Setelah dilakukan perancangan, implementasi, dan pengujian pada penelitian ini, maka dapat diambil beberapa kesimpulan yaitu:

1. Aplikasi *Event Management* berbasis website memiliki fitur penjadwalan anggota panitia berjalan 100% berdasarkan pengujian *alpha* menggunakan metode *black box* testing.
2. Implementasi Particle Swarm Optimization untuk menghasilkan penjadwalan anggota panitia pada aplikasi *Event Menegement* ini dapat diimplementasikan dengan baik. Dalam Particle Swarm Optimization ini, partikel merepresentasikan jadwal *event*, sedangkan dimensi merepresentasikan divisi anggota, dan posisi partikel merepresentasikan anggota. Pada penelitian ini nilai *inertia* yang optimal 0,9 sedangkan nilai pembelajaran kognitif, dan sosial yaitu 1,4. Dalam Percobaannya Nilai rata-rata iterasi saat berhenti 118,6.

### 5.2. Saran

Setelah mendapatkan hasil pengujian yang telah dilakukan oleh penulis, maka untuk penelitian selanjutnya penulis menyarankan menggunakan algoritma Genetika.

## REFERENSI

- [1] A. Noor, Manajemen Event, Bandung : Alfabeta, 2013.
- [2] Norton, Philippa, and J. Nevan Wright. Management of Event Operations. Events Management. Taylor & Francis, 2005.
- [3] Marbun, Yuniar, N. Nikentari, and Martaleli Bettiza. "Perbandingan Algoritma Genetika dan Particle Swarm Optimization dalam Optimasi Penjadwalan Matakuliah." Fak. Tek. Umr (2013): 1-7.
- [4] Zhao, Shuang, Xianli Lu, and Xuejun Li. "Quality of service-based particle swarm optimization scheduling in cloud computing." Proceedings of the 4th International Conference on Computer Engineering and Networks. Springer, Cham, 2015.
- [5] Pinedo, Michael. Scheduling. Vol. 29. New York: Springer, 2012.
- [6] Pudyastuti, Widdhi Purwo. Optimasi Preventive Maintenance Dengan Pso (Particle Swarm Optimization) Pada Semi Lean Solution Pump 107-Jc Di Pabrik I Pt. Petrokimia Gresik. Diss. Institut Teknologi Sepuluh Nopember Surabaya, 2014.
- [7] Juneja, Mudita, and S. K. Nagar. "Particle swarm optimization algorithm and its parameters: A review." 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM). IEEE, 2016..
- [8] Marini, Federico, and Beata Walczak. "Particle swarm optimization (PSO). A tutorial." *Chemometrics and Intelligent Laboratory Systems* 149 (2015): 153-165.
- [9] Handayati, Sahlya, Danang Triantoro Murdiansyah, and Z. K. A. Baizal. "Optimasi Kapasitas Penyimpanan Stego-image Menggunakan Algoritma Particle Swarm Optimization." *eProceedings of Engineering* 3.3 (2016).



- [10] Santosa, Budi. "Tutorial particle swarm optimization." *Sukolilo Surabaya: Kampus ITS 66* (2006)
- [11] Hassan, Rania. "Particle swarm optimization: Method and applications." Presentation at <http://ocw.mit.edu> (2004).
- [12] Widiastuti, Nur Aeni, Stefanus Santosa, and Catur Supriyanto. "Algoritma Klasifikasi data mining naïve bayes berbasis Particle Swarm Optimization untuk deteksi penyakit jantung." *Pseudocode 1.1* (2014): 11-14.
- [13] Mansur, Mansur, Toni Prahasto, and Farikhin Farikhin. "Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi." *JSINBIS (Jurnal Sistem Informasi Bisnis) 4.1* (2014): 11-19.

