

BAB III
PENJADWALAN MATA KULIAH DI DEPARTEMEN PENDIDIKAN
MATEMATIKA FPMIPA UPI DENGAN MENGGUNAKAN
ALGORITMA *PARTICLE SWARM OPTIMIZATION*

Dalam bab ini akan dibahas tentang data yang digunakan dalam penelitian ini, serta model optimisasi dan tahapan algoritma *Particle Swarm Optimization* dalam masalah penjadwalan di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia.

3.1 Analisis Data

Data penelitian diperoleh dari penjadwalan kuliah di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia semester genap tahun akademik 2019-2020 yang terdiri dari data mata kuliah, dosen, kelas, ruangan, hari, dan jam.

3.1.1. Data Mata Kuliah

Terdapat dua program studi di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia, yaitu program studi pendidikan matematika dan program studi matematika. Berdasarkan program studi, mata kuliah dibagi menjadi mata kuliah program studi pendidikan matematika dan mata kuliah program studi matematika. Berdasarkan departemen, mata kuliah dibedakan menjadi mata kuliah keahlian dan mata kuliah dasar umum. Mata kuliah keahlian terdiri dari mata kuliah wajib dan mata kuliah pilihan. Berdasarkan bobotnya, mata kuliah dibagi menjadi dua sks, tiga sks, dan empat sks. Mata kuliah dua sks dan tiga sks dilaksanakan satu kali dalam seminggu, sedangkan mata kuliah empat sks dilaksanakan dua kali dalam seminggu dan dibagi menjadi dua sks tiap satu kali pertemuan.

3.1.2. Data Dosen

Data dosen di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia dibagi menjadi dua, yaitu dosen departemen dan dosen luar departemen. Dosen departemen adalah dosen yang terdaftar sebagai pegawai negeri

sipil yang unit kerjanya di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia, sedangkan dosen luar departemen yaitu dosen Universitas Pendidikan Indonesia yang unit kerjanya di luar Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia. Dosen departemen biasanya mengajarkan mata kuliah keahlian di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia, sedangkan dosen luar departemen mengajarkan mata kuliah dasar umum.

3.1.3. Data Kelas

Program studi di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia terdiri dari program studi pendidikan matematika dan program studi matematika. Program studi pendidikan matematika mempunyai dua kelas, yaitu kelas A dan kelas B. Sedangkan program studi matematika hanya mempunyai satu kelas, yaitu kelas C. Namun untuk angkatan 2019, program studi matematika mempunyai dua kelas, yaitu kelas C dan kelas D. Khusus mata kuliah pembelajaran, kelas A dan B dibagi menjadi kelas A1, A2, B1, dan B2. Terdapat pula kelas gabungan AB untuk mata kuliah pilihan.

3.1.4. Data Ruangan

Ruangan yang tersedia di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia berjumlah 15 ruangan, yaitu S101, S301, S302, S303, S306, E210, E301, E305, E306, E405, dan E406 berada di gedung FPMIPA A, B301 dan B303 berada di gedung FPMIPA B, C301 berada di gedung FPMIPA C, dan Lapangan.

3.1.5. Data Hari

Perkuliahan di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia dilaksanakan pada hari Senin sampai Jumat sehingga jumlah hari perkuliahan dalam seminggu adalah lima hari.

3.1.6. Data Jam

Perkuliahan di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia dilaksanakan pada pukul 07.00 sampai 12.00 WIB dan dilanjutkan pada pukul 13.00 sampai 18.00 WIB. Pukul 12.00 sampai 13.00 digunakan untuk istirahat, salat, dan makan.

3.2 Model Penjadwalan Mata Kuliah

Pada subbab ini dibangun model optimisasi masalah penjadwalan mata kuliah di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia dengan mendefinisikan himpunan dan parameter, serta menentukan variabel keputusan, fungsi tujuan, dan fungsi kendala.

3.2.1. Pendefinisian Himpunan

Berikut adalah definisi himpunan yang akan digunakan dalam pemodelan.

- A : himpunan dosen
- B : himpunan kelas
- C : himpunan mata kuliah
- C_1 : himpunan mata kuliah berbobot empat sks
- D : himpunan ruangan
- E : himpunan hari
- F : himpunan jam

3.2.2. Pendefinisian Parameter

Parameter yang digunakan dalam model optimisasi didefinisikan sebagai berikut.

- $q_{im} = \begin{cases} 1, & \text{jika dosen } i \text{ maksimal mengajar tiga mata kuliah} \\ & \text{pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$
- $s_{im} = \begin{cases} 1, & \text{jika dosen } i \text{ maksimal mengajar dua mata kuliah} \\ & \text{berurutan pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$
- $w_{jm} = \begin{cases} 1, & \text{jika kelas } j \text{ maksimal belajar tiga mata kuliah} \\ & \text{pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$

- $z_{jm} = \begin{cases} 1, & \text{jika kelas } j \text{ maksimal belajar dua mata kuliah berurutan} \\ & \text{pada hari } m \\ 0, & \text{yang lainnya} \end{cases}$

3.2.3. Penentuan Variabel Keputusan

Variabel keputusan dari model optimisasi didefinisikan sebagai berikut.

$$x_{ijklmn} = \begin{cases} 1, & \text{jika dosen } i \text{ mengajar kelas } j \text{ pada mata kuliah } k \\ & \text{di ruangan } l \text{ pada hari } m \text{ dan jam ke } n \\ 0, & \text{yang lainnya} \end{cases}$$

3.2.4. Penentuan Fungsi Kendala

Kendala dari model optimisasi masalah penjadwalan mata kuliah mewakili *hard constrain*, yaitu kendala yang harus dipenuhi pada saat penyusunan jadwal mata kuliah. Kendala-kendala tersebut adalah sebagai berikut.

1. Setiap dosen hanya boleh mengajar satu kelas dalam satu mata kuliah di ruangan pada hari dan jam yang sama

$$\sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} = 1, \forall i \in A$$

2. Setiap mata kuliah diajarkan oleh minimal satu dosen

$$\sum_{i \in A} \sum_{j \in B} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} \geq 1, \forall k \in C$$

3. Setiap kelas tidak boleh dijadwalkan lebih dari satu mata kuliah pada hari dan jam yang sama

$$\sum_{i \in A} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} \leq 1, \forall j \in B$$

4. Setiap ruangan tidak boleh dijadwalkan lebih dari satu mata kuliah pada hari dan jam yang sama

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} x_{ijklmn} \leq 1, \forall l \in D, \forall m \in E, \forall n \in F$$

5. Mata kuliah yang berbobot empat sks dilaksanakan dua kali pertemuan dalam satu minggu

$$\sum_{i \in A} \sum_{j \in B} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} = 2, \forall k \in C_1$$

3.2.5. Penentuan Fungsi Tujuan

Untuk mendefinisikan fungsi tujuan dari model optimisasi ditentukan *soft constrain*, yaitu kendala yang sedapat mungkin dipenuhi oleh model, tetapi jika tidak dipenuhi tidak akan melanggar solusi optimal. Kendala-kendala tersebut adalah sebagai berikut.

1. Setiap dosen maksimal mengajar tiga mata kuliah dalam satu hari

Misal

$$p_{im} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{n \in F} x_{ijklmn}$$

dengan

$$q_{im} = \begin{cases} 1, & \text{jika } p_{im} \leq 3 \\ 0, & \text{yang lainnya} \end{cases}$$

Maka

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} q_{im} x_{ijklmn}$$

2. Setiap dosen maksimal mengajar dua mata kuliah berurutan dalam satu hari

Misal

$$r_{im} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{n \in F} x_{ijklmn}$$

dengan

$$s_{im} = \begin{cases} 1, & \text{jika } r_{im} \leq 2 \\ 0, & \text{yang lainnya} \end{cases}$$

Maka

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} s_{im} x_{ijklmn}$$

3. Setiap mata kuliah yang dilaksanakan dalam dua pertemuan per minggu jarak idealnya minimal dua hari

Misal

$$T = \{t \in E | t = 1, 2\}$$

Maka

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in T} \sum_{n \in F} x_{ijklmn} \cdot x_{ijkl(m+2)n}$$

4. Setiap kelas maksimal belajar tiga mata kuliah dalam satu hari

Misal

$$v_{jm} \sum_{i \in A} \sum_{k \in C} \sum_{l \in D} \sum_{n \in F} x_{ijklmn}$$

dengan

$$w_{jm} = \begin{cases} 1, & \text{jika } v_{jm} \leq 3 \\ 0, & \text{yang lainnya} \end{cases}$$

Maka

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} w_{jm} x_{ijklmn}$$

5. Setiap kelas maksimal belajar dua mata kuliah berurutan dalam satu hari

Misal

$$y_{jm} \sum_{i \in A} \sum_{k \in C} \sum_{l \in D} \sum_{n \in F} x_{ijklmn}$$

dengan

$$z_{jm} = \begin{cases} 1, & \text{jika } y_{jm} \leq 2 \\ 0, & \text{yang lainnya} \end{cases}$$

Maka

$$\sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} z_{jm} x_{ijklmn}$$

3.2.6. Model Optimisasi

Berdasarkan uraian di atas diperoleh model optimisasi masalah penjadwalan mata kuliah di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia adalah sebagai berikut.

Memaksimumkan:

$$z = \sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} (q_{im} + s_{im} + w_{jm} + z_{jm}) x_{ijklmn} \\ + \sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in T} \sum_{n \in F} x_{ijklmn} \cdot x_{ijkl(m+2)n}$$

terhadap

$$\sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} = 1, \forall i \in A$$

$$\begin{aligned} \sum_{i \in A} \sum_{j \in B} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} &\geq 1, \forall k \in C \\ \sum_{i \in A} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} &\leq 1, \forall j \in B \\ \sum_{i \in A} \sum_{j \in B} \sum_{k \in C} x_{ijklmn} &\leq 1, \forall l \in D, \forall m \in E, \forall n \in F \\ \sum_{i \in A} \sum_{j \in B} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} x_{ijklmn} &= 2, \forall k \in C_1 \\ x_{ijklmn} &\in \{0,1\}, \forall i \in A, \forall j \in B, \forall k \in C, \forall l \in D, \forall m \in E, \forall n \in F \end{aligned}$$

3.3 Algoritma PSO untuk Penjadwalan Mata Kuliah

Dalam penelitian ini digunakan algoritma PSO untuk menyelesaikan masalah penjadwalan mata kuliah di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia. Pada algoritma PSO, penjadwalan direpresentasikan sebagai partikel. Terdapat empat partikel yang dilibatkan dalam proses penjadwalan. Partikel pertama mewakili mata kuliah yang diajarkan oleh dosen pada kelas tertentu. Sedangkan partikel ke dua mewakili hari, partikel ke tiga mewakili jam, dan partikel ke empat mewakili ruangan.

Untuk partikel satu terlebih dahulu ditentukan dosen yang mengajar mata kuliah pada kelas tertentu. Oleh karena itu, penjadwalan yang dibahas hanya sebatas menentukan hari, jam, dan ruangan untuk partikel satu. Subbab berikut adalah tahapan algoritma PSO dalam menyelesaikan masalah penjadwalan mata kuliah.

3.3.1. Inisialisasi dan pembangkitan kecepatan dan posisi awal partikel

Sebelum diinisialisasi dan dibangkitkan kecepatan dan posisi awal partikel, data yang akan digunakan dilabeli terlebih dahulu. Contoh pelabelan untuk data hari di Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia seperti pada Tabel 3.1.

Tabel 3.1 Pelabelan Hari

ID_Hari	Hari
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat

Pelabelan untuk data mata kuliah, dosen, kelas, ruangan, dan jam dilakukan seperti pelabelan data hari. Selanjutnya ditentukan batas atas dan batas bawah untuk partikel hari, jam, dan ruangan seperti pada Tabel 3.2.

Tabel 3.2 Batas Atas dan Batas Bawah Partikel Hari, Jam, dan Ruangan

Batas Bawah/Batas Atas	Hari	Jam	Ruangan
Xmin	1	1	1
Xmax	5	12	15

Setelah ditentukan batas atas dan batas bawah, langkah selanjutnya membangkitkan kecepatan dan posisi awal partikel secara acak. Pembangkitan kecepatan dan posisi awal partikel dilakukan dengan persamaan (1) dan (2).

$$x_0^i = x_{min} + rand(x_{max} - x_{min}) \quad (1)$$

$$v_0^i = x_{min} + rand(x_{max} - x_{min}) \quad (2)$$

Keterangan:

x_0^i : posisi awal data ke-i

v_0^i : kecepatan awal data ke-i

x_{max} : batas atas

x_{min} : batas bawah

$rand$: nilai *random* antara 0 dan 1

Misalkan setelah dibangkitkan diperoleh hasil pembangkitan seperti pada Tabel 3.3.

Tabel 3.3 Pembangkitan Posisi Partikel pada Iterasi Pertama

No.	Partikel 1			Partikel 2	Partikel 3	Partikel 4
	Mata Kuliah	Dosen	Kelas	Hari	Jam	Ruangan
1	33	1	1	1	8	9
2	11	33	1	1	2	7
3	19	12	1	1	1	15
4	32	13	1	1	10	5
5	19	9	2	1	7	9
6	32	52	2	1	11	13
7	34	8	3	1	10	15
8	33	54	5	1	6	14
9	16	30	6	1	6	8
10	18	45	6	1	1	5
11	44	43	7	1	12	3
12	48	49	9	1	12	9
13	37	9	12	1	11	9
14	49	21	12	1	12	4
15	36	45	15	1	6	12
16	47	17	18	1	10	4
17	2	46	20	1	9	3
18	52	4	21	1	10	9
19	14	36	1	2	3	2
20	41	10	1	2	8	13
21	3	37	3	2	12	5
22	46	49	4	2	8	3
23	39	35	6	2	12	8
24	16	24	9	2	8	3
25	21	27	9	2	5	13
26	44	44	11	2	9	11
27	35	34	12	2	9	1
28	6	51	12	2	4	9
29	31	26	12	2	7	1
30	40	12	13	2	9	15
31	40	15	14	2	12	7
32	12	2	15	2	1	4
33	49	7	15	2	6	4
34	25	42	15	2	8	3
35	30	17	16	2	3	14
36	22	25	1	3	11	1
37	33	23	2	3	7	2
38	41	31	2	3	8	11
39	22	20	2	3	10	4

40	23	47	3	3	9	14
41	15	18	3	3	8	3
42	15	28	4	3	4	9
43	48	7	6	3	8	2
44	13	41	6	3	8	8
45	18	45	6	3	3	9
46	21	14	6	3	11	12
47	18	2	9	3	12	11
48	18	2	9	3	7	5
49	27	53	13	3	10	5
50	9	16	13	3	3	1
51	10	37	14	3	7	11
52	38	35	15	3	5	9
53	7	31	17	3	12	6
54	50	31	17	3	12	14
55	14	32	2	4	2	4
56	11	55	2	4	6	14
57	3	37	3	4	4	10
58	46	19	3	4	4	7
59	3	21	4	4	7	10
60	1	40	5	4	6	3
61	32	38	5	4	1	12
62	39	6	9	4	5	15
63	44	16	10	4	3	5
64	42	31	12	4	8	7
65	27	16	14	4	8	1
66	29	29	15	4	6	14
67	17	39	15	4	10	9
68	38	35	16	4	4	4
69	26	46	16	4	9	12
70	5	21	20	4	2	1
71	51	51	21	4	5	2
72	45	10	22	4	7	3
73	20	26	23	4	7	3
74	1	3	1	5	2	13
75	1	22	2	5	5	4
76	23	17	4	5	9	14
77	3	21	4	5	2	3
78	34	50	4	5	7	13
79	44	48	8	5	8	7
80	13	15	9	5	7	10
81	24	18	12	5	3	8
82	10	11	13	5	10	11

83	9	43	14	5	1	13
84	19	9	16	5	7	7
85	10	24	16	5	6	11
86	43	5	17	5	2	4
87	8	34	17	5	3	2
88	28	33	19	5	9	2
89	4	37	20	5	6	9

Hasil pembangkitan partikel tersebut kemudian direpresentasikan dalam bentuk vektor. Sebagai contoh, untuk kelas 1 mata kuliah 14 dosen 36 diperoleh hasil pembangkitan pada hari 1 jam 4 di ruangan 6 sehingga posisi awal untuk data kesatu adalah $x_0^1 = (2,3,2)$. Kecepatan pada iterasi pertama diinisialisasi sama dengan posisi awal, yaitu $v_0^1 = (2,3,2)$. Perhitungan tersebut diulangi sebanyak jumlah data. Pada penelitian ini, digunakan data sebanyak 89 data yang diperoleh dengan memasangkan mata kuliah dan kelas dengan dosen yang mengajar.

3.3.2. Menentukan nilai *fitness*

Dalam penelitian ini, fungsi *fitness* menentukan banyaknya *soft constrain* yang dipenuhi. Ketika terjadi pelanggaran terhadap *soft constrain* maka nilai pelanggaran akan ditotalkan untuk satu iterasi. Semakin kecil jumlah pelanggaran yang terjadi, maka nilai *fitness* yang dihasilkan semakin besar. Sebaliknya, semakin besar jumlah pelanggaran yang terjadi, maka nilai *fitness* yang dihasilkan semakin kecil. Pada penelitian ini, fungsi *fitness* yang digunakan adalah sebagai berikut.

Memaksimumkan:

$$f = \sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in E} \sum_{n \in F} (q_{im} + s_{im} + w_{jm} + z_{jm}) x_{ijklmn} \\ + \sum_{i \in A} \sum_{j \in B} \sum_{k \in C} \sum_{l \in D} \sum_{m \in T} \sum_{n \in F} x_{ijklmn} \cdot x_{ijkl(m+2)n}$$

3.3.3. Menentukan nilai *local best* dan *global best*

Local best merupakan posisi terbaik yang pernah dicapai partikel dengan membandingkan nilai *fitness* pada posisi partikel sekarang dengan sebelumnya. *Local best* dipersiapkan untuk mendapatkan solusi terbaik (Cholissodin & Riyandani, 2016). Pada iterasi pertama, *local best* diperoleh dari posisi awal

partikel tersebut. Pada iterasi selanjutnya, *local best* diperoleh dengan membandingkan antara posisi sekarang dan *local best*. Jika posisi sekarang lebih besar dari *local best*, maka *local best* akan di-*update* dengan nilai posisi sekarang.

Global best merupakan posisi terbaik partikel yang diperoleh dengan membandingkan nilai *fitness* terbaik dari keseluruhan partikel dalam *swarm* (Cholissodin & Riyandani, 2016). Pada iterasi pertama, *global best* diperoleh dari nilai *fitness* terbesar. Pada iterasi selanjutnya, *global best* diperoleh dengan membandingkan antara posisi sekarang dan *global best*. Jika posisi sekarang lebih besar dari *global best*, maka *global best* akan di-*update* dengan nilai posisi global sekarang.

3.3.4. Update kecepatan dan posisi partikel

Tahap selanjutnya yaitu meng-*update* kecepatan dan posisi partikel dengan persamaan (3) dan (4). Dalam proses *update* kecepatan dibutuhkan parameter yaitu *inertia factor* (w), *self confidence* (c_1), dan *swarm confidence* (c_2). *Inertia factor* digunakan untuk mengontrol dampak dari perubahan kecepatan. Nilai w memiliki rentang 0,4 – 0,9. Sedangkan *self confidence* dan *swarm confidence* digunakan untuk mengontrol sejauh mana partikel berpindah dalam satu iterasi. Nilai c_1 dan c_2 bernilai sama yaitu dalam rentang 0 – 4. Hasil *update* posisi partikel digunakan untuk menentukan solusi baru yang lebih baik.

$$v_{k+1}^i = w * v_k^i + c_1 * rand * (p^i - x_k^i) + c_2 * rand * (p_k^g - x_k^i) \quad (3)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (4)$$

Keterangan:

w : *inertia factor*

v_k^i : kecepatan sekarang

x_k^i : posisi sekarang

c_1, c_2 : *self confidence* dan *swarm confidence*

rand : nilai *random* antara 0 dan 1

p^i : *local best*

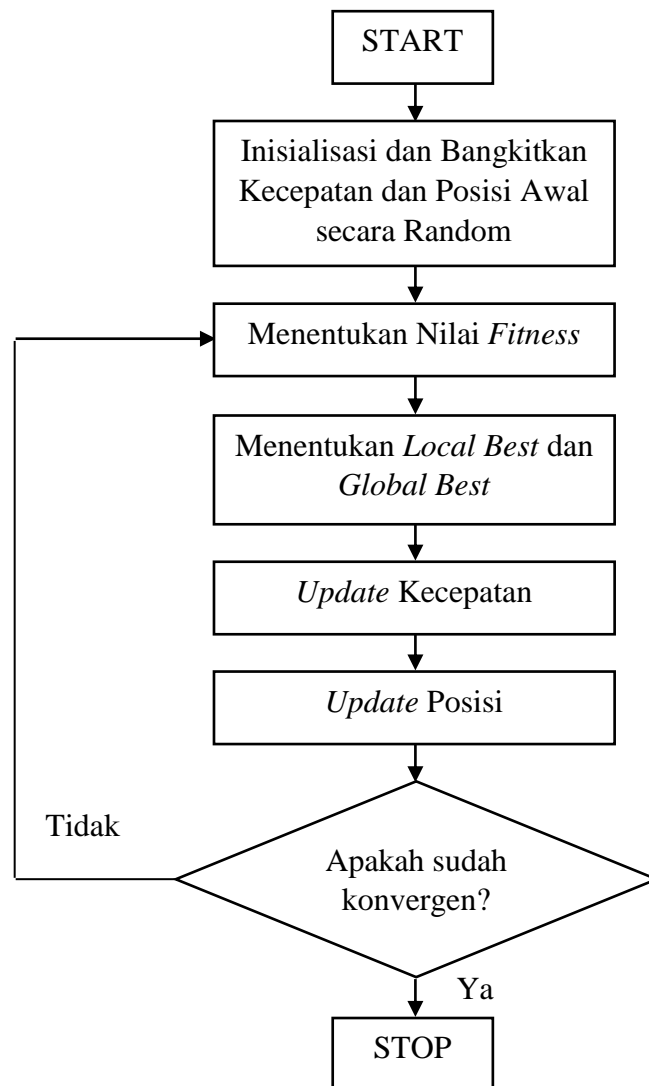
p_k^g : *global best*

x_{k+1}^i : posisi yang dicari

v_{k+1}^i : kecepatan yang dicari

3.3.5. Pengulangan iterasi

Proses iterasi dilakukan sampai ditemukan solusi yang optimal dengan memerhatikan posisi semua partikel. Jika nilai *fitness* sudah menuju ke satu nilai yang sama atau sudah konvergen, maka dapat dikatakan solusi sudah optimal. Jika belum konvergen, maka ulangi Tahap 3.3.2. Selengkapnya, tahapan PSO digambarkan pada Gambar 3.1.



Gambar 3.1 *Flowchart* Algoritma PSO