

Optimasi Pembagian Tugas Dosen Pengampu Mata Kuliah Dengan Metode *Particle Swarm Optimization*

Muhammad Abduh¹, Rekyan Regasari Mardi Putri², Lailil Muflikhah³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹officialabduh@gmail.com, ²rekyan.rmp@ub.ac.id, ³lailil@ub.ac.id

Abstrak

Dalam penyusunan jadwal suatu perkuliahan, hal mutlak yang harus dilakukan adalah pembagian tugas mengajar tenaga kerja sesuai dengan bidang yang mereka tekuni supaya terciptanya kegiatan belajar mengajar yang efektif. Pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang proses penugasan masih dilakukan dengan cara manual dimana pengerjaannya membutuhkan waktu yang tidak sedikit, sehingga dibutuhkanlah suatu metode optimasi yang tepat dalam menangani hal ini. Permasalahan ini dapat diselesaikan dengan metode *heuristic* berbasis populasi, *Particle Swarm Optimization* (PSO) yang telah diterapkan pada berbagai bidang seperti penjadwalan dan penugasan. Data yang digunakan pada penelitian ini merupakan data pembagian tugas dosen pengampu mata kuliah yang berupa prioritas keminatan mengajar dosen terhadap suatu mata kuliah. Berdasarkan hasil perhitungan, dilakukanlah pengujian untuk mengetahui pengaruh parameter uji terhadap nilai *fitness* yang dihasilkan. Dari hasil pengujian parameter PSO, diperoleh jumlah partikel terbaik sebesar 100, jumlah iterasi terbaik 100, dan kombinasi parameter kecepatan $c1$ dan $c2$ sebesar 1.5 dan 1.5 dengan nilai *fitness* yang didapat sebesar 94878. Dari hasil sistem, solusi yang didapat memberikan hasil yang baik, yaitu masih dalam batas toleransi dengan semakin kecilnya nilai kesalahan yang didapat dalam menempatkan dosen terhadap mata kuliah yang sesuai dengan keminatannya.

Kata Kunci: *penugasan, Particle Swarm Optimization, dosen pengampu, keminatan mengajar, mata kuliah, nilai fitness.*

Abstract

In forming a lectures schedule, an absolute thing that had to be done is division of labor teaching duties in accordance with their field in order to create an effective teaching and learning activities. At Faculty of Computer Science (FILKOM) Brawijaya University, the assignment process is still manually designed where it requires some substantial time, therefore it needed a right optimization methods in dealing with this case. This problem can be solved by a population-based heuristic methods, Particle Swarm Optimization (PSO) which has been applied in various fields such as scheduling and assignments. The data used in this study is the data division of lecturers teaching tasks as a priority of lecturer's teaching interests to a course. Through the obtained results, it had tested to find the effect of tested parameters on the resulted fitness values. From PSO parameters test results, it obtained the best particle number as 100, best iteration number as 100, and combination of velocity parameter $c1$ and $c2$ as 1.5 and 1.5 with resulted fitness value as 94878. From the results of system, the obtained assignment solution gives good results, which is still within the limits of tolerance with decreasingly obtained error values in putting a lecturer on courses that according to their interests.

Keywords: *assignments, Particle Swarm Optimization, lecturer, teaching interests, courses, fitness values.*

1. PENDAHULUAN

Dalam sebuah kegiatan belajar-mengajar, salah satu hal pertama yang perlu diperhatikan adalah penjadwalan. Penyusunan jadwal ini tentu tidak dapat dilakukan sembarangan, karena

keberhasilan kegiatan belajar-mengajar yang efektif ditentukan oleh proses penjadwalan yang optimal. Dalam penyusunan jadwal, hal mutlak yang harus dilakukan adalah pembagian tugas mengajar para tenaga kerja sesuai dengan bidang yang mereka tekuni dan mereka inginkan sebab pekerjaan yang sesuai dengan kemampuan

tenaga kerja akan memerlukan nilai *cost* yang lebih sedikit dibandingkan dengan yang tidak sesuai (Taha, 2007). Seringkali proses pembagian tugas/penugasan ini memakan waktu yang tidak sebentar dikarenakan prosesnya yang rumit, mengingat banyaknya tenaga kerja, mata kuliah, dan ruang mengajar. Salah satu lembaga pendidikan yang menemui kendala pada objek ini adalah Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang.

Pada awal semester, pihak akademik akan membagikan angket untuk diisi oleh setiap dosen atau pengajar di FILKOM untuk diisi mengenai keminatan mengajar mata kuliah yang akan mereka ambil pada semester tersebut. Data pada angket yang telah diisi oleh para dosen akan diproses oleh pihak akademik untuk dibuat pembagian tugas mengajar mata kuliah. Pembagian tugas mengajar ini dilihat dari pilihan mata kuliah mayor dan minor, serta urutan prioritas mata kuliah yang diajarkan. Pembagian tugas mengajar tentunya harus memperhatikan aturan-aturan dan batasan-batasan dari setiap parameternya. Seperti jumlah minimal dan maksimal dari SKS maupun mata kuliah yang dapat diambil. Selama ini, pihak akademik menggunakan software *Microsoft Excel* dalam penyusunannya, sehingga apabila terjadi ketidaksesuaian, pengkoreksian pun dilakukan dengan cara manual pula. Disamping aspek-aspek tersebut, banyak cara yang dapat digunakan untuk memperoleh proses pembagian tugas terbaik. Salah satu cara yang dapat digunakan yaitu dengan menggunakan proses optimasi dengan metode *Particle Swarm Optimization* (PSO).

Pada penelitian sebelumnya yang dilakukan oleh Wati & Rochman (2013), metode PSO yang digunakan sebagai basis model penjadwalan kuliah otomatis dapat digunakan sebagai alat bantu pembuatan jadwal yang menghasilkan tabel jadwal yang memenuhi semua kriteria yang bersifat wajib dan dapat mengakomodasi preferensi dosen dalam memilih waktu mengajar. Hal ini dilihat dari hasil yang telah memenuhi *hard constraint* dan meminimalkan *soft constraint* yang dimiliki oleh posisi setiap mata kuliah dengan solusi yang terletak pada optimum global. Sedangkan pada penelitian oleh Hetari et al. (2016), algoritma *hybrid Genetic Algorithm-Simulated Annealing* (GA-SA) dapat digunakan untuk menyelesaikan masalah pembagian tugas yang sama, tetapi nilai *cost* yang didapat masih tergolong tinggi dikarenakan jumlah data yang digunakan

tergolong sedikit. Pada penelitian tugas akhir ini, permasalahan pembagian tugas mengajar diselesaikan dengan metode *Particle Swarm Optimization* (PSO).

PSO adalah sebuah metode *heuristic* berbasis populasi yang dikembangkan oleh Kennedy dan Eberhart pada tahun 1995 yang terinspirasi dari pergerakan kolektif organisme seperti kawanan burung dan kumpulan ikan untuk mensimulasi sikap mereka dalam mencari sumber makanan (Bansal et al., 2011). Salah satu keunggulan PSO dari metode optimasi lain adalah metode ini lebih cepat untuk mencapai konvergensi. Kelebihan yang lain yaitu PSO mudah diimplementasikan, dan memiliki sedikit fungsi operasi dan parameter yang harus ditentukan (Wati & Rochman, 2013). Pada implementasi PSO ini, kandidat solusi merepresentasikan posisi sebuah mata kuliah pada suatu tabel jadwal dimana setiap kandidat solusi memiliki nilai biaya. Populasi partikel dibangkitkan di awal iterasi pertama, lalu partikel-partikel tersebut memperbaiki posisinya menuju posisi terbaik di setiap iterasi, sehingga didapatlah posisi optimum sebuah mata kuliah pada tabel jadwal. Proses ini dilakukan untuk setiap mata kuliah, sehingga diperoleh sebuah jadwal yang utuh. Dengan metode PSO ini, diharapkan didapatlah proses penjadwalan yang singkat dengan nilai akurasi yang memuaskan dan nilai *error* seminimal mungkin.

Berdasarkan uraian diatas, penulis mengajukan penelitian yang berjudul Optimasi Pembagian Tugas Pengajaran Mata Kuliah Dosen Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang Dengan Metode *Particle Swarm Optimization*. Dengan penelitian ini, diharapkan dapat menyelesaikan permasalahan pembagian tugas pengajaran dosen di FILKOM Universitas Brawijaya Malang.

2. LANDASAN KEPUSTAKAAN

2.1 Pembagian Tugas/Penugasan (*Assignment*)

Masalah penugasan (*assignment problem*) adalah suatu masalah mengenai pengaturan objek untuk melaksanakan tugas, dengan tujuan meminimalkan biaya, waktu, jarak, dan sebagainya ataupun memaksimalkan keuntungan (Soemartojo, 1997). Sedangkan menurut Taha (2007), deskripsi yang tepat mengenai model penugasan (*assignment model*)

ialah orang terbaik untuk melakukan sebuah pekerjaan. Hal ini dilihat bahwa pekerja yang memiliki kemampuan yang sesuai dengan pekerjaannya akan memerlukan *cost* yang lebih sedikit dibandingkan dengan yang tidak sesuai, sehingga tujuan dari model penugasan ini adalah untuk menentukan *cost* minimum penugasan pekerja untuk sebuah pekerjaan.

Dalam Agustini dan Rahmadi (2004), dijelaskan bahwa masalah penugasan dalam keadaan tertentu akan mengalami cacah penugasan, dimana jumlah $m \neq n$. Namun, menurut Paendong dan Prang (2011) masalah ini dapat diatasi dengan cara menambahkan *dummy worker* atau *dummy job*.

Pembagian tugas pengajaran mata kuliah merupakan proses menyusun sejumlah komponen yang terdiri atas dosen, mata kuliah, dan jumlah ruangan dengan tujuan tidak terjadi suatu ketidaksinambungan yang menyebabkan terganggunya proses belajar mengajar dan dengan memperhatikan batasan-batasan tertentu. Beberapa batasan yang diberikan oleh tiap-tiap dosen di FILKOM Universitas Brawijaya Malang adalah sebagai berikut:

1. Beban total SKS yang diambil adalah minimal 12 SKS dan maksimal 30 SKS.
2. Jumlah total mata kuliah yang dapat diambil adalah minimal 1 mata kuliah dan maksimal 4 mata kuliah.

2.2 Particle Swarm Optimization (PSO)

Menurut Hsieh et al. (2011), *Particle Swarm Optimization* (PSO) adalah sebuah metode *heuristic* berbasis populasi yang dikembangkan oleh Kennedy dan Eberhart pada tahun 1995. Algoritma ini terinspirasi dari pergerakan kolektif organisme seperti kawanan burung dan kumpulan ikan untuk mensimulasi sikap mereka dalam mencari sumber makanan (Bratton & Kennedy, 2007). Menurut Marini & Walczak (2015) *Swarm* adalah sebuah populasi yang homogen, agen sederhana yang melakukan tugas-tugas dasar dan berinteraksi secara lokal diantara mereka sendiri, dan lingkungan mereka, tanpa disertai adanya pusat kontrol yang memunculkan perilaku kolektif sebagai konsekuensi dari interaksi organisasi dan lokal (dan sebagian *stochastic*). Sedangkan menurut Fletcher (2000) dalam bidang sains, optimasi adalah proses yang merujuk pada pengidentifikasian elemen yang terbaik (menurut kriteria yang telah ditentukan oleh pengguna) diantara kemungkinan set-set

alternatif yang tersedia. Sedangkan dalam bidang matematika, hal ini biasanya dilakukan dengan mendefinisikan tujuan dalam hal fungsi parameter f , sehingga optimasi bermuara untuk menemukan nilai-nilai parameter yang meminimalkan atau memaksimalkan (tergantung pada tugas tertentu) dari parameter f itu sendiri.

Menurut Marini & Walczak (2015), dalam PSO setiap kandidat solusi disebut sebagai “partikel” dan merepresentasi titik dalam ruang D -dimensional, dengan D adalah jumlah parameter yang akan dioptimasi. Ruang dimensional ini seringkali digambarkan sebagai sebuah matriks 2 dimensi dimana baris menggambarkan jumlah partikel dan kolom menggambarkan jumlah kromosom. Untuk posisi dari partikel ke- i dapat dideskripsikan dengan vector x_i :

$$x_i = [x_{i1} \ x_{i2} \ x_{i3} \ \dots + x_{iD}] \tag{1}$$

dan populasi dari kandidat solusi N merupakan *swarm*:

$$x = \{x_1, x_2, x_3, x_n\} \tag{2}$$

Dalam mencari solusi optimal dari suatu masalah, partikel mendefinisikan lintasan dalam ruang parameter berdasarkan ekuasi:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{3}$$

dimana t dan $t + 1$ mengindikasikan dua iterasi berturut-turut algoritma dan v_i adalah vector komponen-kecepatan dari partikel i sepanjang dimensi D . Vektor kecepatan ini mengatur cara partikel bergerak melintasi ruang pencarian dan terdiri dari kontribusi tiga buah kondisi: kondisi pertama, yang didefinisikan inersia atau momentum yang mencegah partikel mengubah arah secara drastis dengan melacak kecepatan sebelumnya; kondisi kedua yang disebut sebagai komponen kognitif, menyeimbang kecenderungan partikel untuk kembali ke posisi terbaik yang ditemukan sebelumnya; kondisi ketiga, yang disebut komponen sosial, dengan mengidentifikasi kecenderungan partikel untuk bergerak ke arah posisi terbaik dari seluruh *swarm* (atau lingkungan lokal partikel, tergantung pada apakah yang diimplementasikan adalah PSO global atau parsial). Berdasarkan pertimbangan ini, kecepatan didefinisikan sebagai:

$$v_i^{(t+1)} = v_i^{(t)} + c_1(p_i - x_i^{(t)})R_1 + c_2(g - x_i^{(t)})R_2 \tag{4}$$

dimana p_i , yang disebut sebagai “personal best”

partikel, yaitu koordinat solusi terbaik yang diperoleh sejauh ini oleh individu tertentu, sedangkan g adalah “global best”, yaitu keseluruhan solusi terbaik yang diperoleh dengan *swarm*. Untuk konstanta akselerasi c_1 dan c_2 yang bernilai real dan biasanya dikisaran $0 \leq c_1, c_2 \leq 4$, disebut “koefisien kognitif” dan “koefisien sosial”, dan memodulasi besarnya langkah-langkah yang diambil oleh partikel di dalam arah menuju personal best dan global best masing-masing. Sedangkan R_1 dan R_2 adalah dua matriks diagonal dari nomor acak yang dihasilkan dari distribusi seragam di $[0, 1]$, sehingga komponen sosial dan kognitif memiliki pengaruh stochastic pada aturan perubahan kecepatan pada ekuasi tersebut. Dengan demikian, lintasan ditarik oleh partikel yang semi-acak karena mereka berasal dari kontribusi tarik sistematis terhadap solusi personal best dan global best dan pembobotan stochastic dari kedua kondisi akselerasi ini.

Menurut Marini & Walczak (2015), secara umum langkah-langkah dari algoritma PSO diketahui sebagai berikut:

1. Inisialisasi. Untuk setiap N partikel:
 - a. Inisialisasi posisi

$$x_i(0) \forall i \in 1 : N \tag{5}$$
 - b. Inisialisasi posisi terbaik partikel sampai posisi awalnya

$$p_i^{(0)} = x_i^{(0)} \tag{6}$$
 - c. Hitung nilai *fitness* dari masing-masing partikel dan bila $f(x_j(0)) \geq f(x_i(0)) \forall i \neq j$ maka inisialisasi global best adalah

$$g = x_i^{(0)} \tag{7}$$
2. Sampai kriteria terpenuhi, ulangi langkah berikut:
 - a. Ubah kecepatan partikel sesuai dengan ekuasi pada Persamaan 4.
 - b. Ubah posisi partikel sesuai dengan ekuasi pada Persamaan 3.
 - c. Evaluasi nilai *fitness* partikel

$$f(x_i^{(t+1)}) \tag{8}$$
 - d. Apabila $f(x_i(t+1)) \geq f(p_i)$, ubah personal best:

$$p_i = x_i^{(t+1)} \tag{9}$$
 - e. Apabila $f(x_i(t+1)) \geq f(g)$, ubah global best:

$$g = x_i^{(t+1)} \tag{10}$$

3. Pada akhir proses iterasi, solusi terbaik dijadikan sebagai variable g .

Pada tahun 1998, terdapat perubahan metode PSO pada proses perhitungan kecepatan, yaitu diperkenalkannya variabel bobot inersia (Eberhart & Shi, 1998). Variabel ini ditambahkan kedalam rumus kecepatan yang sudah ada seperti:

$$v_i^{(t+1)} = w \cdot v_i^{(t)} + c_1(p_i - x_i^{(t)})R_1 + c_2(g - x_i^{(t)})R_2 \tag{11}$$

dengan w sebagai variabel bobot inersia. Konsep penggunaan variabel bobot inersia ini digunakan untuk pengaturan eksplorasi dan eksploitasi algoritma dengan lebih baik. Hasilnya terbukti dengan meningkatnya kinerja pada beberapa penerapan metode ini. Seleksi bobot inersia yang sesuai menghasilkan eksplorasi global dan lokal yang seimbang dengan hasil rata-rata jumlah iterasi yang lebih sedikit dalam menemukan solusi optimal yang dapat diterima. Pada awal perkembangannya, nilai bobot inersia berkurang secara berkala dari sekitar 0.9 sampai 0.4. Beberapa penyesuaian dilakukan agar sistem yang dibuat dapat melacak dan mengoptimasi secara dinamis sehingga dibuatlah suatu rumus untuk nilai bobot inersia seperti

$$w = 0.5 + \left(\frac{random}{2}\right) \tag{12}$$

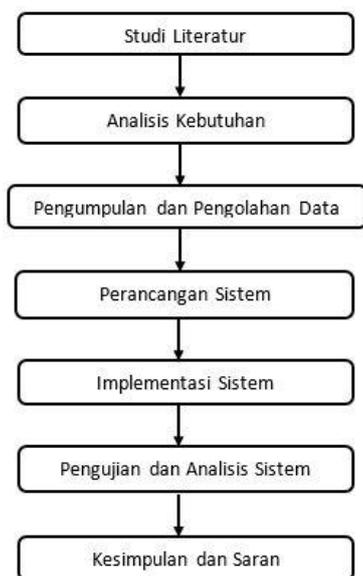
dengan menggunakan bilangan *random*. Rumus ini menghasilkan bilangan *random* dengan variasi antara 0.5 sampai dengan 1.0 dengan rata-rata 0.75 (Eberhart & Shi, 1998).

Pada PSO, kondisi perhentian dapat ditentukan dari berbagai cara; seperti batas nilai *fitness* yang telah ditetapkan telah terpenuhi, sebagian besar partikel mencapai titik yang sama dalam suatu ruang pencarian atau nilai *fitness* yang dihasilkan tidak mengalami perubahan yang signifikan.

3. METODOLOGI

Bab ini menjelaskan metode yang digunakan serta langkah - langkah yang dilakukan dalam pembuatan sistem optimasi pembagian tugas dosen pengampu dengan metode *Particle Swam Optimization*. Metodologi penelitian yang dilakukan dalam penelitian ini terdiri dari beberapa tahap diantaranya adalah studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi sistem, pengujian sistem, dan

penarikan kesimpulan. Langkah - langkah dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Contoh Penggunaan Gambar

3.1 Studi Literatur

Studi literatur dari berbagai sumber diperlukan untuk merealisasikan optimasi penugasan dosen pengampu dengan metode PSO. Informasi serta pustaka yang memiliki keterkaitan dengan penelitian ini diperoleh dari jurnal, situs internet, dosen pembimbing, serta rekan-rekan mahasiswa.

3.2 Analisis Kebutuhan

Analisis kebutuhan dalam pembangunan sistem optimasi penugasan dosen pengampu dengan menggunakan metode *Particle Swarm Optimization* digunakan untuk menganalisis kebutuhan apa saja yang diperlukan dalam pembangunan sistem ini sehingga dapat mencapai tujuan yang diinginkan. Adapun kebutuhan yang didapat pada tahapan ini dibagi menjadi dua, yaitu Kebutuhan Fungsional, dan Kebutuhan Data.

3.3 Pengumpulan dan Pengolahan Data

Dari analisis kebutuhan yang telah ditentukan, dilakukanlah pengumpulan data dari narasumber, yaitu pihak akademik yang bertanggung jawab dalam melakukan pembagian tugas mengajar para dosen FILKOM tiap semesternya. Namun karena ketidakcukupan jumlah data yang didapat, maka digunakanlah data *dummy* untuk menentukan prioritas mata kuliah dosen yang dibuat

berdasarkan format data asli. Data *dummy* yang digunakan dapat digunakan sebagai tolak ukur dalam pembuatan sistem, sehingga apabila dikemudian hari data asli diterapkan, sistem dapat lebih mudah beradaptasi. Data yang digunakan akan disimpan ke dalam *database* untuk memudahkan sistem dalam mengambil (*fetch*) data dan menyimpan *output* sistem nantinya.

3.4 Perancangan Sistem

Perancangan sistem dilakukan dengan cara memecah proses sistem dari awal hingga akhir untuk mempermudah proses implementasi dan pengujian. Dari proses pemecahan tersebut didapatkanlah perancangan langkah kerja dari sistem secara menyeluruh yang dibagi menjadi tiga, yaitu *input* yang dimasukkan, proses sistem, dan *output* yang dihasilkan. Hasil dari perancangan sistem ini disusun ke dalam diagram model yang menjelaskan tentang langkah kerja sistem secara terstruktur.

3.5 Implementasi Sistem

Implementasi *interface* menggunakan bahasa pemrograman *Java* dengan *software NetBeans IDE 7.2*. Untuk data penelitian yang digunakan akan disimpan didalam *dataset MySQL* dengan menggunakan *localhost XAMPP*.

3.6 Pengujian dan Analisis Sistem

Pengujian dilakukan untuk mengetahui hubungan nilai *fitness* terhadap parameter yang diujikan. Parameter yang baik akan didapat apabila nilai *fitness* yang didapat bernilai tinggi. Pengujian yang akan dilakukan adalah pengujian jumlah partikel, pengujian jumlah iterasi, dan pengujian parameter kecepatan *c1* dan *c2*.

3.7 Kesimpulan dan Saran

Berdasarkan hasil pada langkah-langkah penelitian yang telah dilakukan, maka ditariklah kesimpulan dan saran. Kesimpulan akan menjadi jawaban dari rumusan masalah yang telah dibuat pada Bab 1. Sedangkan Saran merupakan harapan penulis yang dimaksudkan untuk memperbaiki kesalahan yang terjadi serta memberikan pertimbangan untuk penelitian dan pengembangan kedepannya.

4 PERANCANGAN

4.1 Formulasi Permasalahan

Permasalahan yang dihadapi adalah pembagian tugas mengajar dosen terhadap mata kuliah yang dibuka yang dilakukan oleh pihak akademik dengan melihat data angket yang telah diisi oleh masing-masing dosen. Pada data angket tersebut terdapat beberapa parameter seperti prioritas mata kuliah yang akan diambil mulai dari prioritas satu sampai prioritas lima dengan keterangan prioritas satu merupakan prioritas yang paling diutamakan. Parameter selanjutnya adalah mata kuliah mayor dan minor yang menentukan mata kuliah mana yang lebih diprioritaskan untuk diajar oleh dosen. Sifat mata kuliah mayor dan minor ini mirip dengan parameter prioritas mata kuliah, dimana prioritas mata kuliah yang termasuk jenis mayor lebih diutamakan. Setiap dosen dapat mengisi angket tersebut sesuai dengan mata kuliah yang ingin diajarkan dengan memperhatikan batasan-batasan yang ada seperti setiap dosen hanya boleh mengambil mata kuliah yang ingin diajarkan dengan ketentuan untuk jumlah SKS, minimal adalah 12 dan maksimal adalah 25; dan jumlah mata kuliah, minimal adalah 1 dan maksimal adalah 4. Berdasarkan data yang telah diisi para dosen, maka pihak akademik akan menentukan mata kuliah apa saja yang sebaiknya diambil oleh setiap dosen. Namun untuk saat ini proses pembagian tugas mengajar ini masih dikerjakan dengan menggunakan *Microsoft Excel*, sehingga apabila terjadi ketidaksesuaian diperlukan proses yang cukup menyita waktu untuk membenahinya dikarenakan proses kerjanya yang masih *semi-manual*.

4.2 Siklus Penyelesaian Masalah Menggunakan *Particle Swarm Optimization*

Data yang digunakan pada penelitian ini adalah data *dummy* berupa pemetaan mengajar dosen semester genap 2015/2016 prodi S1 Informatika/Ilmu Komputer di FILKOM Universitas Brawijaya Malang. Berikut contoh potongan data pemetaan dosen pada Tabel 1 dan data mata kuliah yang dibuka dapat dilihat pada Tabel 2.

Tabel 1. Tabel Pemetaan Data Dosen

Nama Dosen	Kode Dosen	Prioritas Mengajar	Mata Kuliah	
			Mayor	Minor
Dosen A	Dosen1	1, 6, 5	1, 6	5
Dosen B	Dosen2	1, 7, 6, 4	1, 7, 6	4
Dosen C	Dosen3	7, 1, 5, 6	7, 1	5, 6
Dosen D	Dosen4	2, 8, 1	2, 8	1
Dosen E	Dosen5	5, 9, 7	5, 9	7
Dosen F	Dosen6	1, 9, 6, 3	1, 9, 6	3
Dosen G	Dosen7	3, 4, 8, 2	3, 4	8, 2

Tabel 2. Tabel Mata Kuliah

No.	Mata Kuliah	SKS	Kelas
1	Analisis dan Perancangan Sistem	3	5
2	Jaringan Komputer	4	11
3	Sistem Terdistribusi	3	5
4	Basis Data Terdistribusi	3	8
5	Desain dan Analisis Algoritma	3	6
6	Mixed Reality	3	4
7	Pemrograman Web	4	9
8	Pengenalan Pola	3	5
9	Sistem Pakar	3	7

Siklus penyelesaian terhadap masalah yang diteliti dengan menggunakan PSO dimulai dari:

1. Inisialisasi Partikel
Inisialisasi partikel digunakan untuk menentukan posisi awal partikel. Inisialisasi partikel dilakukan dengan cara membangkitkan matriks bilangan bulat secara acak dengan *range* 1-20, sesuai dengan jumlah dosen yang digunakan yaitu 20. Panjang kromosom partikel disini sesuai dengan jumlah kelas yang dibuka, sedangkan jumlah partikel dapat disesuaikan dengan *input* yang diberikan.
2. Membangkitkan Matriks Biaya
Matriks biaya digunakan dalam menentukan salah satu *cost* untuk mencari nilai *fitness*, yaitu *cost* keminatan. *Cost* keminatan adalah pelanggaran kromosom partikel terhadap data keminatan dosen. Pada Tabel 1, terdapat prioritas mengajar dosen beserta mata kuliah mayor dan minor yang akan diambil. Mata kuliah mayor adalah mata kuliah yang lebih diprioritaskan daripada mata kuliah minor. Dari sini dibuatlah suatu

nilai pembobotan terhadap prioritas tersebut pada Tabel 3.

Tabel 3. Tabel Pembobotan

Prioritas Mata Kuliah	Jenis Mata Kuliah	
	Mayor	Minor
Prioritas ke - 1	1	2
Prioritas ke - 2	3	4
Prioritas ke - 3	5	6
Prioritas ke - 4	7	8
Prioritas ke - 5	9	10
Bukan prioritas	100	

Dari nilai pembobotan yang telah ditentukan maka dibuatlah matriks biaya. Sebagai contoh, pada Dosen1, *cost* terhadap mata kuliah Analisis dan Perancangan Sistem (APS) merupakan 1, karena mata kuliah APS merupakan mata kuliah prioritas pertama yang ingin beliau ajar sekaligus termasuk mata kuliah mayor. Lalu untuk *cost* terhadap mata kuliah Mixed Reality (MIXER) merupakan 3, karena MIXER merupakan mata kuliah prioritas kedua dan termasuk jenis mata kuliah mayor, begitu pula seterusnya. Sedangkan untuk mata kuliah yang tidak diambil oleh dosen, maka akan diberi nilai 100 untuk memberikan nilai *cost* yang cukup besar apabila nilai kromosom yang terdapat dalam partikel memberikan nilai yang kurang cocok dengan data yang ada. Potongan matriks biaya dapat dilihat pada Tabel 4.

Tabel 4. Tabel Matriks Biaya

Kode Mata Kuliah	Dosen						
	1	2	3	4	5	6	7
APS_1	1	1	3	5	10	1	10
APS_2	1	1	3	5	10	1	10
APS_3	1	1	3	5	10	1	10
JARKOM_1	10	10	10	1	10	10	8
JARKOM_2	10	10	10	1	10	10	8
JARKOM_3	10	10	10	1	10	10	8
JARKOM_4	10	10	10	1	10	10	8
SISTER_1	10	10	10	10	10	8	1
SISTER_2	10	10	10	10	10	8	1
SISTER_3	10	10	10	10	10	8	1
BDT_1	10	8	10	10	10	10	3
BDT_2	10	8	10	10	10	10	3

DAA_1	6	10	6	10	1	10	10
DAA_2	6	10	6	10	1	10	10
DAA_3	6	10	6	10	1	10	10
DAA_4	6	10	6	10	1	10	10
DAA_5	6	10	6	10	1	10	10
MIXER_1	3	5	8	10	10	5	10
MIXER_2	3	5	8	10	10	5	10
MIXER_3	3	5	8	10	10	5	10

3. Menghitung Nilai *Fitness*

Setiap partikel memiliki nilai *fitness* yang didapat dengan cara menghitung total dari ketiga *cost*, yaitu *cost* keminatan, *cost* SKS, dan *cost* kelas. *Cost* keminatan didapat dari proses matriks biaya. Untuk *cost* SKS didapat dengan cara melihat total SKS yang diambil oleh dosen tersebut. Apabila total SKS yang diambil kurang dari 12, maka *cost*-nya adalah dengan mengurangi 12 dengan total SKSnya. Tetapi apabila total *cost*-nya lebih dari 25, maka *cost*-nya adalah dengan mengurangi total SKS dengan 25. Sedangkan untuk *cost* kelas, didapat dengan cara melihat total jumlah mata kuliah yang diambil oleh dosen tersebut. Apabila total mata kuliah yang diambil kurang dari 1, maka *cost*-nya adalah 1. Apabila total mata kuliah yang diambil lebih dari 4, maka *cost* didapat dengan cara mengurangi total mata kuliah dengan 4. Untuk rumus *fitness* dapat dilihat pada persamaan berikut:

$$fitness = C - total_cost \tag{13}$$

Konstanta yang digunakan disini bernilai 100000, karena bilangan tersebut mudah dikurangkan dengan nilai total *cost* yang didapat dari tiap partikel yang berkisar sampai puluhan ribu. Selain itu hasil pengurangan dari konstanta dapat digunakan sebagai nilai *fitness* yang umumnya memandang nilai yang lebih besar sebagai nilai yang lebih baik.

4. Menentukan Partikel *pbest* dan *gbest*

Partikel *pbest* merupakan partikel yang terbaik, dengan kata lain memiliki nilai *fitness* yang paling besar dibandingkan dengan partikel-partikel disekitarnya. Menentukan partikel *pbest* dilakukan dengan cara membandingkan nilai *fitness* tiap partikel pada suatu iterasi. Partikel *gbest* adalah partikel terbaik dari seluruh iterasi.

Menentukan partikel *gbest* dilakukan dengan cara membandingkan partikel-partikel *pbest* sehingga didapatlah partikel dengan nilai *fitness* tertinggi.

5. Menghitung Nilai Kecepatan
Kecepatan merupakan fungsi PSO untuk menentukan arah posisi baru partikel. Tiap kromosom partikel dicari nilai kecepatannya dengan menggunakan rumus pada Persamaan 4. Untuk nilai *w* yang digunakan disini bernilai 0.75 berdasarkan percobaan pada Eberhart & Shi (1998), sedangkan nilai *c1* dan *c2* dapat ditentukan melalui *input* yang diberikan.
6. Menghitung Nilai Posisi Baru
Menghitung nilai posisi baru atau *update* posisi partikel digunakan untuk menentukan posisi partikel pada iterasi selanjutnya dengan melakukan penjumlahan nilai kecepatan yang telah dilakukan dengan nilai kromosom partikel pada posisi sebelumnya. Partikel pada posisi baru ini akan dicari kembali nilai *fitness*-nya dan dibandingkan dengan nilai *fitness* partikel pada posisi sebelumnya untuk mencari partikel *pbest* dan *gbest*.

5 PENGUJIAN DAN ANALISIS

5.1 Pengujian Jumlah Partikel

Pengujian jumlah partikel dilakukan untuk melihat pengaruh jumlah partikel yang digunakan terhadap nilai *fitness*. Jumlah partikel yang digunakan adalah 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 yang dilakukan masing-masing 10 kali untuk dicari rata-rata nilai *fitness* pada setiap pengujian. Jumlah iterasi yang digunakan adalah 10 dengan nilai parameter *c1* = 1 dan *c2* = 1. Dari data hasil pengujian dibuatlah grafik untuk melihat pengaruhnya terhadap nilai *fitness* seperti pada Gambar 2.



Gambar 2. Grafik Hasil Pengujian Jumlah Partikel

Dari hasil pengujian jumlah partikel, nilai

fitness terendah ada pada jumlah partikel terkecil yaitu 10 dengan nilai rata-rata *fitness* 87187.7 dan tertinggi ada pada partikel dengan jumlah 100 dengan nilai rata-rata *fitness* 89248.8. Dari Gambar 2 bisa disimpulkan bahwa semakin banyak jumlah partikel yang digunakan, maka nilai *fitness* yang didapat akan semakin tinggi. Peningkatan nilai *fitness* terjadi karena semakin banyak partikel, maka semakin besar ruang lingkup pencarian solusi dan variasi nilai *fitness* yang didapat. Banyaknya variasi nilai *fitness* membuat konvergensi dini dapat dihindari.

5.2 Pengujian Jumlah Iterasi

Pengujian jumlah iterasi dilakukan untuk melihat pengaruh jumlah iterasi terhadap nilai *fitness*. Jumlah iterasi yang digunakan disini adalah 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 dengan masing-masing dilakukan sebanyak 10 kali percobaan. Sedangkan untuk jumlah partikel yang digunakan adalah 100 yang didapat dari hasil terbaik dari pengujian jumlah partikel dan untuk nilai parameter *c1* = 1 dan *c2* = 1. Dari data hasil pengujian dibuatlah grafik untuk melihat pengaruhnya terhadap nilai *fitness* seperti pada Gambar 3.

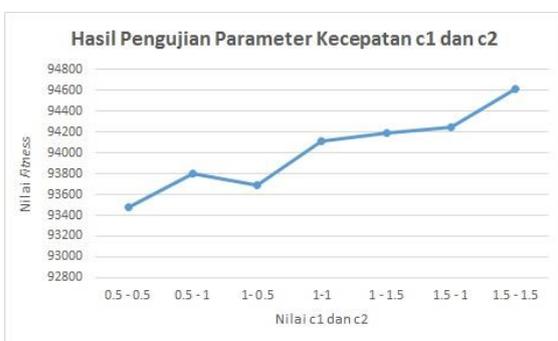


Gambar 3. Grafik Hasil Pengujian Jumlah Iterasi

Dari hasil uji coba jumlah iterasi, nilai *fitness* terendah ada pada jumlah iterasi terkecil yaitu 10 dengan nilai rata-rata *fitness* 89248.8 dan tertinggi ada pada jumlah iterasi 100 dengan nilai rata-rata *fitness* 93924.5. Dari Gambar 3 diatas, dapat dilihat bahwa nilai rata-rata *fitness* mengalami kestabilan. Hal ini disebabkan oleh semakin seringnya proses *update* nilai *fitness* dan pergerakan partikel yang lebih sering terhadap suatu posisi baik yang optimum maupun yang tidak, namun seiring dengan bertambahnya iterasi, kemungkinan posisi optimum yang didapat akan cenderung tinggi sehingga nilai *fitness* yang didapat menjadi lebih baik.

5.3 Pengujian Parameter Kecepatan $c1$ dan $c2$

Pengujian parameter kecepatan $c1$ dan $c2$ dilakukan untuk menemukan kombinasi parameter $c1$ dan $c2$ yang tepat untuk menghasilkan solusi yang terbaik. Kombinasi parameter $c1$ dan $c2$ yang digunakan adalah kombinasi dari bilangan 0.5, 1, 1.5 dan dilakukan sebanyak 10 kali percobaan. Untuk jumlah partikel yang digunakan adalah 100 yang didapat dari hasil terbaik dari pengujian jumlah partikel dan jumlah iterasi yang digunakan adalah 100 sesuai dengan hasil terbaik dari pengujian jumlah iterasi. Dari data hasil pengujian dibuatlah grafik untuk melihat pengaruhnya terhadap nilai *fitness* seperti pada Gambar 4.



Gambar 4. Grafik Hasil Pengujian Parameter Kecepatan $c1$ dan $c2$

Dari hasil uji coba pada Gambar 6.3, nilai *fitness* yang dihasilkan oleh kombinasi parameter $c1$ dan $c2$ mengalami perubahan yang signifikan. Rata-rata nilai *fitness* terendah didapatkan dari kombinasi parameter $c1$ dan $c2$ dengan nilai masing-masing 0.5 dan 0.5, yaitu sebesar 93476.4. Sedangkan untuk rata-rata nilai *fitness* terbaik didapatkan dari kombinasi $c1$ dan $c2$ dengan nilai masing-masing 1.5 dan 1.5, yaitu sebesar 94611.1. Pada kombinasi parameter $c1$ dan $c2$ dengan nilai tinggi, waktu yang diperlukan untuk proses akan semakin lama bila dibandingkan dengan yang lain, namun kemungkinan terjadinya konvergensi dini akan semakin kecil. Pergerakan partikel oleh parameter kecepatan $c1$ dan $c2$ yang merupakan koefisien percepatan terhadap parameter $pbest$ dan $gbest$ sulit diukur seiring dengan besar nilai parameter yang digunakan. Hal ini karena pergerakan partikel PSO yang melihat jejak posisi partikel sebelumnya tidak selalu mendekati posisi optimum.

5.4 Analisis Hasil Optimasi Dengan Metode Particle Swarm Optimization

Dari pengujian jumlah partikel, pengujian jumlah iterasi, pengujian parameter kecepatan $c1$ dan $c2$ didapatkan bahwa nilai parameter yang optimal adalah jumlah partikel sebesar 100, jumlah partikel sebesar 100, dan jumlah kombinasi $c1$ dan $c2$ sebesar 1.5 dan 1.5. Dari nilai parameter seperti yang telah disebutkan didapat nilai *fitness* dengan jumlah tertinggi 94878 dari *range* 83170 sampai dengan 99835 atau sekitar 0.703 dengan *range* 0 sampai dengan 1. Jumlah pelanggaran minimum menghasilkan nilai *fitness* sebesar 99835, yang didapat dari pengurangan nilai konstanta terhadap nilai pelanggaran sebesar 165. Nilai pelanggaran minimum ini didapat dengan menilik nilai *cost* keminatan minimum, yaitu 1 untuk masing-masing 165 kromosom, dan 0 untuk nilai *cost* sks dan *cost* kelas. Sedangkan jumlah pelanggaran maksimumnya menghasilkan nilai *fitness* sebesar 83170. Nilai pelanggaran maksimum ini didapat dari pengurangan nilai konstanta terhadap nilai pelanggaran sebesar 16830, yaitu *cost* keminatan sebesar 100 untuk masing-masing 165 kromosom, 1 nilai *cost* sks dan 1 nilai *cost* kelas untuk masing-masing 165 kromosom. Nilai *fitness* yang dihasilkan ini masih dapat bertambah baik dengan melihat bahwa konvergensi nilai belum terlihat sehingga penambahan jumlah maksimum parameter dapat dilakukan. Namun, penambahan jumlah maksimum parameter ini juga akan menyebabkan waktu perhitungan yang semakin lama dan alokasi memori yang semakin besar sehingga diperlukan spesifikasi mesin dengan memori RAM yang lebih besar dengan alokasi memori yang lebih baik. Banyaknya jumlah nilai parameter uji yang digunakan disini adalah pertimbangan penulis dengan melihat kemampuan perangkat yang digunakan.

Sedangkan untuk hasil *output* dari sistem, nilai *error* yang dihasilkan semakin berkurang seiring bertambahnya jumlah parameter yang digunakan. Dari nilai *fitness* yang dihasilkan, terdapat kesalahan penempatan dosen terhadap mata kuliah dengan rata-rata sebanyak 52 dari 165 mata kuliah. Hal ini dapat diperkecil dengan menambah nilai parameter. Bertambahnya nilai parameter seperti jumlah partikel, menyebabkan semakin luasnya ruang lingkup pencarian sehingga posisi optimum lebih cepat dicapai dan variasi nilai *fitness* yang didapat semakin

banyak. Hal ini menyebabkan berkurangnya terjadi konvergensi dini pada nilai *fitness* sehingga pencarian solusi dapat diteruskan. Bertambahnya jumlah iterasi menyebabkan semakin banyaknya proses *update* nilai *fitness* dilakukan karena pergerakan partikel terhadap suatu posisi lebih sering. Arah grafik nilai *fitness* yang dihasilkan pada bertambahnya jumlah iterasi cenderung lebih tinggi. Hal ini dipengaruhi pergerakan partikel ke arah posisi yang lebih optimum. Sedangkan kombinasi parameter kecepatan *c1* dan *c2* yang lebih baik didapat dari nilai kombinasi yang sama dan jumlah yang lebih besar. Hal ini disebabkan arah pencarian partikel yang semakin mendekati posisi partikel *pbest* dan *gbest* sehingga umumnya solusi optimal akan dicapai lebih cepat seiring semakin tingginya nilai *fitness* yang dihasilkan.

6 KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Pengimplementasian metode *Particle Swarm Optimization* untuk optimasi pembagian tugas dosen pengampu mata kuliah pada penelitian ini memberikan hasil yang cukup baik dengan melihat jumlah kesalahan yang terus menurun dengan semakin bertambahnya jumlah parameter partikel. Meski hasil yang didapat pada penelitian ini masih cukup rendah tetapi apabila ditinjau lebih jauh lagi dari perolehan nilai *fitness* yang semakin baik pada setiap bertambahnya jumlah parameter dan belum terjadinya konvergensi nilai. Hal ini dapat diatasi dengan menambahkan jumlah maksimum parameter uji walau membutuhkan proses perhitungan yang cukup lama dan alokasi memori perangkat yang lebih besar.
2. Dengan melakukan pengujian-pengujian untuk mengetahui pengaruh nilai parameter terhadap nilai *fitness* yang dihasilkan, yaitu pengujian jumlah partikel, pengujian jumlah iterasi, dan pengujian parameter kecepatan *c1* dan *c2*. Berdasarkan pengujian yang telah dilakukan, hasil parameter PSO yang paling optimal adalah dengan menggunakan jumlah partikel sebesar 100, jumlah iterasi 100, dan nilai parameter *c1* dan *c2* masing-masing sebesar 1.5 dan 1.5 dengan nilai *fitness* sebesar 94878.

Saran untuk pengembangan terhadap penelitian ini adalah:

1. Nilai *fitness* masih dapat ditingkatkan dengan menambah jumlah maksimum parameter pengujian walaupun membutuhkan waktu komputasi yang lebih lama dan alokasi memori perangkat yang lebih banyak, sehingga akan lebih baik apabila menggunakan perangkat keras dengan spesifikasi yang lebih besar untuk pengalokasian memori yang lebih baik.
2. Nilai *cost* yang didapat dari penggunaan metode PSO ini masih tergolong tinggi, sehingga untuk masalah penugasan, metode PSO ini dapat digabungkan dengan metode optimasi yang lain.
3. Menambah data keminatan, sehingga kesalahan penempatan mata kuliah pada perhitungan dapat diminimalisir dengan semakin banyaknya data.

DAFTAR PUSTAKA

- Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S. & Abraham, A., 2011. *Inertia Weight Strategies In Particle Swarm Optimization*. Third World Congress on Nature and Biologically Inspired Computing 2011.
- Eberhart, R. C., Shi, Y., 2001. *Particle Swarm Optimization: Developments, Applications, Resources*. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on Vol. 1, Mei 2001.
- Herjanto, E., 2001. *Manajemen Produksi dan Operasi (Edisi Revisi)*. Jakarta. PT. Gramedia.
- Hetari, R., Cholissodin, I., Mahmudy, W. F., 2016. *Implementasi Hybrid Genetic Algorithm dan Simulated Annealing untuk Pembagian Tugas Mengajar Dosen PTIIK*. Repositori Jurnal Mahasiswa PTIIK UB Vol. 7, No. 15.
- Hsieh, H. I., Lee T. P., Lee, T. S., 2011. *A Hybrid Particle Swarm Optimization and Support Vector Regression Model for Financial Time Series Forecasting*. International Journal of Business Administration Vol.2, No. 2; May 2011.
- Mahmudy, W. F., 2006. *Penerapan Algoritma Genetika Pada Optimasi Model Penugasan*. Natural Vol.10, No. 3, p. 197-207.

- Mansur., Prahasto, T., Farikhin., 2014. *Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi*. Jurnal Sistem Informasi Bisnis 01 (2014).
- Marini, F., Walczak, B., 2015. *Particle Swarm Optimization (PSO). A Tutorial*. Chemometrics and Intelligent Laboratory Systems.
- Paendong, M., Prang, J. D., 2011. *Optimasi Pembagian Tugas Karyawan Menggunakan Metode Hungarian*. Jurnal Ilmiah Sains Vol. 11 No.1, April 2011.
- Poli, R., Kennedy, J., Blackwell, T., 2007. *Particle Swarm Optimization An Overview*. Springer Science + Business Media, LLC 2007.
- Rachman, R. A., Syarif, D., Sari, R. P., 2012. *Analisa dan Penerapan Metode Particle Swarm Optimization Pada Optimasi Penjadwalan Kuliah*. Jurnal Teknik Informatika, Vol 1 September 2012.
- Rahajoeningroem, T., Aria, M., *Studi Komparasi Algoritma Particle Swarm Optimization Pada Aplikasi Filter Adaptive Noise Cancellation*. Majalah Ilmiah UNIKOM Vol. 11, No. 1.
- Santosa, B., Willy, P., 2011. *Metoda Metaheuristik, Konsep, dan Implementasi*. Surabaya. Graha Ilmu.
- Taha, H. A., 2007. *Operations Research: An Introduction 8th Edition*. Pearson Prentice Hall. Pearson Education, Inc.
- Wati, D. A. R., Rochman, Y. A., 2013. *Model Penjadwalan Matakuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO)*. Jurnal Rekayasa sistem Industri Vol.2, No.1, 2013.