

Optimasi Pola Penyusunan Barang dalam Peti Kemas Menggunakan Algoritma *Particle Swarm Optimization*

Erny

Abstrak

Penyusunan barang dalam peti kemas adalah salah satu masalah penting untuk beberapa perusahaan. Karena salah satu cara untuk mendapatkan keuntungan yang sebesar-besarnya yaitu dengan mengoptimalkan penyusunan barang. Oleh sebab itu dibutuhkan sebuah algoritma yang dapat menghasilkan penyusunan barang agar penyimpanan barang menjadi optimal. Pada skripsi ini membahas solusi menggunakan algoritma *Particle Swarm Optimization (PSO)*.

Dalam algoritma PSO ini, struktur partikel yang digunakan memberikan informasi berupa data integer berupa nomor barang dan posisi barang, dimana dalam setiap iterasi partikel tersebut di update kecepatan dan posisinya berdasarkan nilai *fitness*. *Fitness* berupa banyaknya barang yang dapat masuk ke dalam peti kemas yang menyisakan paling sedikit ruang kosong. Hasil penelitian menunjukkan bahwa algoritma PSO dapat diterapkan pada permasalahan optimasi penyusunan barang dalam peti kemas.

Kata kunci : penyusunan barang dalam peti kemas, *fitness*, PSO, *Particle Swarm Optimization*, optimasi.

Abstract

Preparation of goods in containers is one of the important problems for some companies. Because one way to get the maximum benefit is to optimize the preparation of goods. Therefore required an algorithm that can be produce the goods preparation in order to be optimal storage of goods. In this paper, discusses a solution using Particle Swarm Optimization (PSO) algorithm.

In the PSO algorithm, the structure of the particles that are used to provide information in the form of an integer data such us number and position of the goods, which is in each iteration the updated particle velocity and position based on fitness value. Fitness in the form of the number of items that can be fit into containers which is leaving the least amount of empty space. Final result shows that PSO algorithm can be implementing on the optimization of the preparation of goods in containers.

Keywords : *preparation of goods in containers, fitness, PSO, Particle Swarm Optimization, optimization.*

1. Pendahuluan

Kemajuan dunia di berbagai bidang pada masa sekarang ini muncul berbagai masalah besar yang harus diatasi dengan cepat dan efisien. Semakin berkembangnya dunia usaha, dituntut pula efisiensi di segala bidang untuk menekan biaya yang dikeluarkan dengan tujuan semakin meningkatkan laba yang di dapat. Melihat kasus tersebut dapat disimpulkan bahwa optimasi mempunyai peran yang sangat penting

dalam dunia usaha untuk menekan biaya pengeluaran seminimal mungkin. (Thiang, 2004)

Untuk memecahkan masalah optimasi tersebut diperlukan suatu algoritma yang baik. Salah satu teknik solusi masalah optimasi yaitu dengan menggunakan teknik *Particle Swarm Optimization (PSO)*. Algoritma *Particle Swarm Optimization (PSO)* merupakan teknik optimasi stokastik berdasarkan populasi yang terinspirasi oleh tingkah laku sosial kawanan burung atau ikan untuk menyelesaikan permasalahan-permasalahan optimasi. (Suyanto, 2004)

Untuk memecahkan masalah optimasi tersebut diperlukan suatu algoritma yang baik. Salah satu teknik solusi masalah optimasi yaitu dengan menggunakan teknik *Particle Swarm Optimization (PSO)*. Algoritma *Particle Swarm Optimization (PSO)* merupakan teknik optimasi stokastik berdasarkan populasi yang terinspirasi oleh tingkah laku sosial kawanan burung atau ikan untuk menyelesaikan permasalahan-permasalahan optimasi. (Suyanto, 2004)

2. Tinjauan Pustaka

2.1 *Particle Swarm Optimization (PSO)*

Particle Swarm Optimization (PSO) adalah teknik optimasi berbasis populasi yang dikembangkan oleh James Kennedy dan Russel Eberhart pada tahun 1995 (J. Kennedy dan R. Eberhart, 1995). Teknik ini terinspirasi oleh tingkah laku sosial pada kawanan burung yang berduyun-duyun (*bird flocking*) atau gerombolan ikan yang berenang berkelompok (*fish schooling*). Kecerdasan seperti inilah yang diadopsi oleh kedua ilmuwan tersebut untuk membangun suatu teknik optimasi, yang mereka namakan *particle swarm optimization* (optimasi kawanan partikel). Mereka menyebutnya partikel karena pada dasarnya banyak makhluk hidup yang memiliki kecerdasan seperti burung dan ikan, misalnya belalang, lebah dan sebagainya (Suyanto, 2010).

PSO memiliki banyak kesamaan dengan teknik-teknik *evolutionary computation* yang lain, seperti *Genetic Algorithms (GA)*, *Evolutionary Strategies (ES)*, *Evolutionary Programming*, dan sebagainya. PSO maupun GA dimulai dengan suatu populasi yang terdiri dari sejumlah individu (yang menyatakan solusi) yang dibangkitkan secara acak dan selanjutnya melakukan pencarian solusi optimum melalui perbaikan individu untuk sejumlah generasi tertentu. Tetapi berbeda dengan GA, PSO tidak menggunakan operator-operator evolusi seperti rekombinasi (*crossover*) dan mutasi. PSO memiliki *memory* untuk menyimpan solusi terbaik, sedangkan GA tidak punya. Setiap partikel pada PSO tidak pernah mati, sedangkan individu pada GA bisa mati dan digantikan dengan individu baru. Pada PSO, posisi dan kecepatan terbang partikel tersebut bisa menghasilkan solusi baru yang lebih baik (Suyanto, 2010).

2.2 Konsep Dasar

Sebagaimana dinyatakan sebelumnya, PSO mensimulasikan perilaku dari sekelompok burung. Dimisalkan ada skenario seperti ini, sekelompok burung sedang terbang secara acak mencari makanan di sebuah area. Hanya terdapat satu potong makanan pada area tersebut. Semua burung tidak mengetahui dimana lokasi makanan yang sebenarnya. Tetapi mereka dapat mengetahui seberapa jauh letak makanan yang ada pada setiap pencarian. Sehingga strategi terbaik yang dapat digunakan untuk mencari makanan adalah dengan cara mengikuti burung yang lokasinya paling dekat dengan makanan tersebut. PSO belajar dari skenario tersebut dan menggunakannya untuk memecahkan masalah optimasi. Pada PSO, satu “burung” menyatakan satu solusi di dalam ruang masalah. Burung tadi direpresentasikan sebagai “partikel” dalam algoritma. Algoritma PSO menggunakan populasi dari sekumpulan partikel, dimana setiap partikel yang ada mewakili sebuah solusi yang mungkin untuk sebuah permasalahan optimasi. Semua partikel memiliki nilai *fitness* yang dievaluasi oleh fungsi *fitness* yang akan dioptimalkan, dan memiliki kecepatan (*velocity*).

Pada setiap iterasi, tiap partikel diperbaharui oleh dua nilai, yang merupakan nilai terbaik. Yang pertama adalah nilai terbaik yang dicapai tiap-tiap partikel, disebut sebagai *pbest*. Nilai yang lain adalah nilai terbaik secara keseluruhan yang dimiliki oleh partikel manapun dalam populasi, nilai ini disebut *gbest*. Setelah menemukan kedua nilai terbaik tersebut partikel memperbaharui kecepatan (*v*) dan posisinya (*x*) dengan persamaan berikut :

$$v_i = (v_i - 1) + (r_1 - r_2) + \dots - \dots \quad (1)$$

$$x_i = (x_i - 1) + (v_i) \quad (2)$$

Dimana :

i = indeks partikel iterasi

V_i = *velocity* partikel ke- i

X_i = posisi partikel ke- i

P_i = posisi terbaik dari partikel ke- i (*pbest*)

P_g = posisi terbaik dari semua partikel (*gbest*)

$c_{1,2}$ = *learning rate*

$r_{1,2}$ = bilangan acak dari distribusi seragam [0,1], $r \in U(0,1)$

2.3 Parameter Algoritma *Particle Swarm Optimization*

Adapun 2 langkah utama ketika menerapkan algoritma *particle swarm optimization* pada masalah optimasi yaitu :

1. Representasi partikel (solusi)
2. Fungsi *fitness*

Tidak terlalu banyak parameter yang dibutuhkan pada algoritma *particle swarm optimization*, ini beberapa parameternya :

1. Jumlah Partikel

Jumlah partikel yang dipilih adalah tergantung pada persoalan yang dihadapi. Jumlah partikel yang umum digunakan berkisar antara 20 sampai 40.

2. *Learning Rate* atau Laju Belajar

c_1 dan c_2 merupakan faktor belajar yang menyatakan seberapa besar partikel itu dihubungkan ke arah solusi yang potensial dari ruang solusi. Pada umumnya nilai-nilai untuk koefisien akselerasi c_1 dan $c_2 = 2.0$.

3. *Inertia Weight* (w)

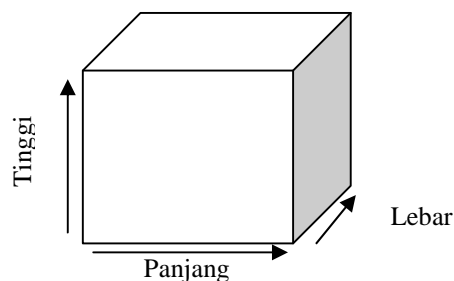
Parameter ini berguna untuk mengontrol keseimbangan antara kemampuan eksplorasi global dan lokal, serta penurunan kecepatan untuk menghindari stagnasi partikel pada optimum lokal.

4. Kondisi Berhenti

Banyak cara untuk membangun kondisi berhenti, diantaranya adalah iterasi dihentikan ketika PSO telah mencapai iterasi maksimum, atau PSO telah menemukan nilai optimum tertentu atau kesalahan minimum yang diinginkan.

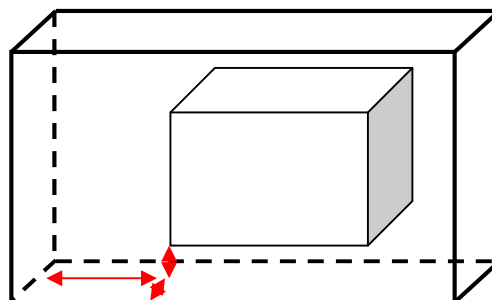
2.5 Pengaturan Tata Letak Barang

Dalam skripsi ini, untuk barang dan peti kemas, panjang akan selalu berasosiasi dengan sumbu X, dan lebar akan selalu berasosiasi dengan sumbu Y dan tinggi akan selalu berasosiasi dengan sumbu Z.



Gambar 2.1 Barang berbentuk kubus

Setiap barang mempunyai tiga properti yaitu panjang (p), lebar (l), dan tinggi (t). Misalkan bahwa p' adalah panjang yang baru pada posisi n , l' adalah lebar yang baru pada posisi n , dan t' adalah tinggi yang baru pada posisi n . Panjang akan selalu berasosiasi dengan sumbu X, lebar akan selalu berasosiasi dengan sumbu Y dan tinggi akan selalu berasosiasi dengan sumbu Z.



2.4 Tujuan Pengaturan Tata Letak Barang

Beberapa tujuan dalam melakukan pengaturan tata letak barang adalah :

1. Menghasilkan susunan barang sesuai dengan kategori tertentu. Misalnya barang yang lebih berat diletakkan di bawah dari barang yang lebih ringan.
2. Menghasilkan aturan barang yang padat yaitu susunan barang yang lebih mementingkan pada optimasi ruang yang digunakan oleh susunan barang tersebut.

Tugas akhir ini akan menggunakan tujuan yang kedua, yaitu untuk menghasilkan susunan barang yang menyisakan sesedikit mungkin ruang kosong pada peti kemas.

3. Pembahasan Hasil

3.1 Penerapan Algoritma PSO Pada Pengaturan Tata Letak Barang

Pada pengaturan tata letak barang menggunakan algoritma PSO, hal yang menjadi solusi adalah partikel dimana partikel ini berisi informasi berupa nomor barang dan posisi barang yang harus dioptimasi. Misalnya sebuah permasalahan tata letak barang yang melibatkan empat buah barang, maka struktur partikel yang digunakan adalah seperti gambar 3.1.

| | | | |
|--------------------|--------------------|--------------------|--------------------|
| Nomor Barang 1 | Nomor Barang 2 | Nomor Barang 3 | Nomor Barang 4 |
| Posisi Barang 1 | Posisi Barang 2 | Posisi Barang 3 | Posisi Barang 4 |

Gambar 3.1 Partikel Barang

Dalam perancangan sistem algoritma PSO, struktur partikel yang digunakan memberikan informasi berupa data integer berupa nomor barang dan posisi barang.

Secara garis besar proses algoritma *Particle Swarm Optimization* dari tiga tahap yaitu pembangkitan posisi serta kecepatan partikel, *update velocity* dan *update* posisi. Model yang akan digunakan untuk permasalahan optimasi peti kemas ini adalah sebagai berikut ;

1. Menentukan nilai dari parameter parameter algoritma *Particle Swarm Optimization* yaitu W , $C1$ dan $C2$.
2. Pembangkitan posisi serta kecepatan awal partikel
Langkah pertama adalah membangkitkan posisi serta awal partikel, maksudnya adalah membuat suatu data barang yang memiliki informasi
3. Evaluasi *fitness*
Nilai *fitness* digunakan untuk mengetahui posisi partikel, baik buruknya posisi partikel dilihat dari nilai fitnessnya Fungsi *fitness* yang digunakan adalah sebagai berikut :

$$= (\quad)$$

4. *Update velocity* (update kecepatan)

Menggunakan persamaan :

$$v(t) = v(t-1) + (p - v(t-1)) + (g - v(t-1)) -$$

5. *Update posisi*

Menggunakan persamaan :

$$p(t) = p(t-1) + v(t)$$

Dalam update posisi ini, partikel atau dalam kasus ini adalah volume 1 barang yang dipilih secara acak dan akan mengalami perubahan sesuai hasil *update*.

6. Kondisi selesai

Terdapat dua kondisi selesai yang dapat menghentikan proses algoritma *Particle Swarm Optimization* yaitu :

- Jika jumlah iterasi maksimum telah tercapai
- Jika *fitness* terbaik telah dicapai atau kesalahan minimum yang diinginkan.

3.2 Uji Coba dan Analisis

Pada bagian ini dilakukan pengukuran kinerja dari optimasi penyusunan barang dalam peti kemas. Dimana yang pertama di definisikan tujuan pengujian, scenario pengujian, parameter yang digunakan dalam pengujian ini. Kemudian hasil pengujian dianalisis untuk menentukan parameter-parameter yang berpengaruh terhadap algoritma PSO sebagai proses optimasi penyusunan barang dalam peti kemas.

3.2.1 Tujuan Pengujian

Berikut adalah tujuan dari pengujian yang dilakukan :

- Menganalisis parameter algoritma *Particle Swarm Optimization* yaitu $C1$ (*factor learning* untuk partikel), $C2$ (*factor learning* untuk swarm) dan W (*inertia weight*).
- Menganalisis hasil optimasi penyusunan barang dalam peti kemas untuk kasus uji barang heterogen.

3.2.2 Skenario Pengujian

Parameter pengujian yang digunakan pada algoritma *Particle Swarm Optimization* yaitu :

- m : jumlah partikel dan nc : jumlah iterasi
Nilai m dan nc yang diuji cobakan adalah $\{10,10\}$
- $C1$: *factor learning* untuk particle
Nilai $C1$ yang diujicobakan adalah $C1 \{0.5, 2\}$
- $C2$: *factor learning* untuk swarm
Nilai $C2$ yang diujicobakan adalah $C2 \{0.5, 2\}$
- W (*inertia weight*)
Nilai W yang diujicobakan adalah $\{0.4, 0.9\}$

Sehingga diperoleh 8 kombinasi parameter untuk diujikan. (ada pada halaman Lampiran).

Berdasarkan scenario penguian di atas maka nanti akan didapat parameter terbaik untuk nilai C1, C2, dan W dimana nilai tersebut digunakan untuk melakukan optimasi penyusunan barang dalam peti kemas. Kemudian dilihat nilai *fitness* mana yang terbaik.

Untuk membantu proses perhitungan barang agar dapat berjalan cepat maka dicari skala pembagi dari ukuran barang dan ukuran peti kemas. Dengan demikian ukuran barang dan ukuran peti kemas akan menjadi lebih kecil sehingga optimasi dapat berjalan lebih cepat.

Data barang yang digunakan adalah seperti pada tabel 3.1 yang terdiri dari 20 barang dengan ukuran barang yang berbeda-beda (heterogen) dengan satuan dm untuk mempermudah hitungan. Dan data peti kemas yang digunakan adalah seperti pada table 3.2.

Tabel 3.1 Data barang untuk kasus heterogen

| Id Barang | Panjang (dm) | Lebar (dm) | Tinggi (dm) | Volume (dm ³) |
|-----------|--------------|------------|-------------|---------------------------|
| 1 | 4 | 15 | 4 | 240 |
| 2 | 15 | 4 | 6 | 360 |
| 3 | 4 | 10 | 4 | 160 |
| 4 | 7 | 15 | 7 | 735 |
| 5 | 5 | 8 | 19 | 760 |
| 6 | 10 | 5 | 6 | 300 |
| 7 | 3 | 4 | 6 | 72 |
| 8 | 14 | 5 | 16 | 1120 |
| 9 | 8 | 14 | 4 | 448 |
| 10 | 4 | 6 | 15 | 360 |
| 11 | 15 | 5 | 4 | 300 |
| 12 | 4 | 20 | 2 | 160 |
| 13 | 13 | 6 | 5 | 390 |
| 14 | 5 | 8 | 10 | 400 |
| 15 | 10 | 7 | 6 | 420 |
| 16 | 4 | 8 | 16 | 512 |
| 17 | 15 | 20 | 3 | 900 |
| 18 | 8 | 4 | 9 | 288 |
| 19 | 15 | 8 | 10 | 1200 |
| 20 | 5 | 10 | 12 | 600 |
| | | | | 9725 |

Tabel 3.2 Data peti kemas untuk kasus barang heterogen

| Panjang (dm) | Lebar (dm) | Tinggi (dm) | Volume (dm ³) |
|--------------|------------|-------------|---------------------------|
| 20 | 20 | 20 | 8000 |

3.2.3 Analisa Hasil Uji Coba Parameter

Untuk uji coba dengan kombinasi parameter 1 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 0.5 | 0.5 | 0.4 | 6568 | 10 |

Dari hasil uji coba dengan menggunakan parameter 1, dimana $C1 = 0.5$, $C2 = 0.5$, dan $w = 0.4$ terlihat bahwa nilai *fitness* yang diperoleh sebesar 6568 dengan iterasi 10 iterasi.

Untuk uji coba dengan kombinasi parameter 2 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 0.5 | 0.5 | 0.9 | 753 | 10 |

Parameter yang digunakan pada uji coba dengan kombinasi parameter 2 yaitu dimana $C1 = 0.5$, $C2 = 0.5$, dan $w = 0.9$. Diperoleh nilai *fitness* sebesar 753 dengan iterasi 10 iterasi.

Untuk uji coba dengan kombinasi parameter 3 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 0.5 | 2 | 0.4 | 54 | 7 |

Pada uji coba dengan kombinasi parameter 3 yang digunakan yaitu $C1 = 0.5$, $C2 = 2$, dan $w = 0.4$. Dan dari hasil uji coba ini diperoleh nilai *fitness* sebesar 54.

Untuk uji coba dengan kombinasi parameter 4 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 0.5 | 2 | 0.9 | 184 | 4 |

Sedangkan untuk uji coba dengan kombinasi parameter 4 yang digunakan yaitu $C1 = 0.5$, $C2 = 2$, dan $w = 0.9$. Terlihat bahwa diperoleh nilai *fitness* sebesar 184 pada iterasi kurang dari 10 yaitu hanya 4 iterasi.

Untuk uji coba dengan kombinasi parameter 5 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 2 | 0.5 | 0.4 | 6608 | 10 |

Pada uji coba 5 dengan kombinasi parameter $C1 = 2$, $C2 = 0.5$, dan $w = 0.4$ diperoleh nilai *fitness* sebesar 6608 pada iterasi maksimal yaitu iterasi ke 10.

Untuk uji coba dengan kombinasi parameter 6 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 2 | 0.5 | 0.9 | 2003 | 10 |

Untuk uji coba dengan kombinasi parameter 6 yaitu $C1 = 2$, $C2 = 0.5$, dan $w = 0.9$. Dari hasil uji coba diperoleh nilai *fitness* sebesar 2003 pada iterasi ke 10.

Untuk uji coba dengan kombinasi parameter 7 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 2 | 2 | 0.4 | 21 | 8 |

Pada uji coba 7 dengan kombinasi parameter $C1 = 2$, $C2 = 2$, dan $w = 0.4$. Dari hasil uji coba ini diperoleh nilai *fitness* sebesar 21 dengan 8 iterasi.

Untuk uji coba dengan kombinasi parameter 8 :

| C1 | C2 | W | fitness | iterasi |
|-----------|-----------|----------|----------------|----------------|
| 2 | 2 | 0.9 | 36 | 4 |

Pada uji coba ke 8 dengan kombinasi parameter diman $C1 = 2$, $C2 = 2$, dan $w = 0.9$. Dari hasil uji coba ini diperoleh nilai *fitness* sebesar 36 dengan iterasi hanya kurang dari 10 iterasi yaitu 4 iterasi.

4. Penutup

Berdasarkan hasil pengujian untuk $m = 10$ dan $nc = 10$ diperoleh parameter $C1$, $C2$, dan W yaitu **{2, 2, 0.9}** dengan nilai *fitness* terbaik yaitu 36. Dimana merupakan sisa ruang kosong di dalam peti kemas. Ini berarti volume barang optimal tersebut yaitu volume kontainer dikurangi sisa ruang kosong (*fitness*). Yaitu **$8000 - 36 = 7964$** .

Dan dari data barang pada tabel 3.1 diketahui bahwa volume total dari fitness ialah 7964 dm^3 sedangkan volume kontainer yaitu $8 \text{ m}^3 = 8000 \text{ dm}^3$. Jadi nilai kualitas rata-rata yang diperoleh sebanyak :

$$\text{---} \times 100\% = 99,5\%$$

Sebaiknya untuk memudahkan proses perhitungan uji coba ke depannya menggunakan sebuah program untuk menyelesaikan permasalahan optimasi menggunakan Algoritma *Particle Swarm Optimization*.

Sebaiknya ada perbaikan lebih lanjut mengenai algoritma PSO ini yang menyangkut simulasi penyusunan barang dalam kontainer baik dalam hal batasan-batasan masalah, maupun parameter-parameternya serta iterasi maksimal yang diujicobakan

Daftar Pustaka

- A. Masoeri, A., dan Indrawan, T. 2010. *Optimasi Distribusi Kapal Perang Armada Timur menggunakan PSO Algoritma*. ITS. Tanggal akses : 7 Februari 2011.
- Kennedy, J. and Eberhart, R. 1995. *Particle Swarm Optimization*, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948. Tanggal akses : 4 November 2010.
- Kurniawati Arik, dan Rifa'i Irfan, Q. H. 2009. *Analisis Penerapan Algoritma Particle Swarm Optimization pada Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika Universitas Trunojoyo Madura*. Madura : Universitas Trunojoyo.
- Nuraisyah, A. 2011. *Simulasi Optimasi Pengisian Kontainer Menggunakan Algoritma Genetika*. Makassar : Universitas Hasanuddin.
- Suyanto. 2010. *Algoritma Optimasi (Deterministik atau Probabilitik)*. Graha Ilmu, Yogyakarta.
- Thiang, dkk. 2004. *Optimasi Pengisian Kontainer dengan Menggunakan Algoritma Genetika*. Surabaya : Universitas Kristen Petra.