Analisis Perbandingan Metode Genetic Algorithm dan Particle Swarm Optimization dalam Menilai Tingkat Optimasi Hasil Pada Bin Packing Problem Satu Dimensi

Adidtya Perdana

Program Studi Teknik Informatika, Sekolah Tinggi Teknik Harapan, Medan, Sumatera Utara Email: adid.dana@gmail.com / adidtya.perdana.stth@gmail.com

Abstract

Bin Packing Problem (BPP) merupakan permasalahan kombinatorial yang termasuk kedalam NP-Complete dalam masalah pemaketan barang-barang yang memiliki ukuran yang berbeda-beda kedalam wadah dengan ukuran yang tetap. Biasanya untuk menyelesaikan BinPacking Problem dapat menggunakan metode First Fit, Next Fit, Best Fit, First Fit Decreasing, Next Fit Decreasing, dan Best Fit Decreasing. Namun metode-metode tersebut masih memiliki kekurangan sehingga dapat digunakan metode metaheuristic yang mampu memberikan hasil yang lebih optimal. Metode-metode metaheuristic yang dapat digunakan dalam menyelesaikan permasalahan Bin Packing Problem adalah Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO). Dalam tulisan ini, penulis akan membahas mengenai perbandingan Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO) dalam menilai tingkat optimasi hasil pada permasalahan Bin Packing Problem satu dimensi serta membandingkannya dengan metode First Fit (FF) dan First Fit Decreasing (FFD) sebagai acuan nilai optimal. Dengan dilakukannya perbandingan ini diharapkan dapat memberikan pilihan metode-metode yang dapat digunakan untuk menyelesaikan permasalahan Bin Packing Problem satu dimensi.

1. Pendahuluan

Bin Packing Problem atau Permasalahan Wadah Pemaketan (BPP) merupakan suatu permasalahan kombinatorial yang menitikberatkan pada pemaketan benda-benda dengan ukuran yang berbeda-beda kedalam suatu wadah dengan ukuran yang sudah ditentukan. Bin Packing Problem membutuhkan solusi terbaik dalam mencari objek-objek (benda-benda) yang optimal yang sesuai dengan wadah yang telah disediakan [1]. Bin Packing Problem merupakan permasalahan yang bersifat NP-complete, dimana permasalahan ini dapat diselesaikan menggunakan metode metaheuristic. Dengan menggunakan metode metaheuristic permasalahan Bin Packing Problem dapat diselesaikan dengan lebih optimal [2].

Genetic Algorithm atau Algoritma Genetika merupakan metode metaheuristic yang dapat digunakan untuk menyelesaikan berbagai permasalahan nondeterministic polynomial (NP) yang dapat memberikan hasil yang optimal. Genetic Algorithm diciptakan oleh John Holland pada tahun 1975 yang sistem kerjanya diambil dari konsep evolusi dimana didalamnya terdapat cara kerja dari seleksi alam dan genetika alam [3].

Selain Genetic Algorithm terdapat metode metaheuristic lainnya yang dapat digunakan untuk optimasi yaitu Particle Swarm Optimization (PSO). Particle Swarm Optimization mengambil konsep sifat dan perilaku sekelompok makhluk hidup, seperti sekawanan burung atau ikan, yang sedang mencari makan. Ketika sekawanan makhluk hidup ini mencari makan, mereka bergerak bersamaan tanpa ada yang mengkoordinasi, tetapi pergerakannya memiliki derajat keteraturan tertentu. Metode ini diciptakan oleh Eberhert dan Kennedy pada tahun 1995 dan sempai sekarang sudah banyak dikembangkan oleh banyak peneliti [4].

Pada dasarnya *Bin Packing Problem* memiliki algoritma atau metode khusus yang dapat digunakan antara lain *First Fit*, *Next Fit*, dan *Best Fit*. Ketiga metode ini selanjutnya dikembangkan lagi dengan menambah teknik pengurutan sehingga menghasilkan metode baru yaitu *First Fit Decreasing*, *Next Fit Decreasing*, dan *Best Fit Decreasing* [1]. Keenam metode ini mampu menyelesaikan permasalahan *Bin*

Packing dengan cukup baik, namun masih memiliki beberapa kekurangan. Dengan digunakannya metode metaheuristic seperti Genetic Algorithm (GA) dan Particle Swarm Optimization (PSO) diharapkan dapat menyelesaikan permasalahan Bin Packing Problem dan memberikan hasil yang lebih optimal.

Berdasarkan penjelasan diatas, penelitian ini akan dibahas perbandingan metode *Genetic Algorithm* (GA) dan *Particle Swarm Optimization* (PSO) dalam menyelesaikan permasalahan *Bin Packing Problem* satu dimensi dengan acuan hasil optimal dari metode *First Fit* (FF) dan *First Fit Decreasing* (FFD). Dikarenakan metode GA dan PSO menggunakan pembangkit bilangan acak kemungkinan hasil yang ditampilkan akan berbeda pada setiap percobaan. Dan hasil tersebut akan dibandingkan lagi dengan hasil dari FF dan FFD.

2. Tinjauan Pustaka

2.1 Bin Packing Problem

Packing Problem merupakan suatu permasalahan kombinatorial (combinatorial problem) termasuk kedalam NP-Complete deterministic polynomial complete) yang membahas masalah bagaimana memaketkan barang-barang atau benda-benda yang ukurannya berbeda-beda kedalam suatu wadah atau bin yang memiliki ukuran yang tetap (tidak berubah). Sebagai contoh, diberikan sejumlah n benda dengan ukuran s_1 , s_2 , s_3 ,..., s_n , dan diberikan sejumlah bin atau wadah dengan ukuran m. Setiap benda harus ditempatkan kedalam bin sehingga besar total semua benda tidak melebihi setiap bin dan memberikan jumlah bin yang minimum. Jika ditulis kedalam bentuk persamaan:

$$\begin{aligned} & minimize \quad z = \sum_{i=1}^n yi \\ & Subjek \; dari \quad \sum_{i=1}^n w_j x_{ij} \leq c y_i, \; i \in \mathbb{N} = \{1, \dots, n\} \\ & \sum_{i=1}^n x_{ij} = 1, \quad j \in \mathbb{N}, \\ & y_i = 0 \; or \; 1, \quad i \in \mathbb{N}, \\ & x_{ij} = 0 \; or \; 1, \quad i \in \mathbb{N}, \end{aligned}$$

Dapat disimpulkan pada *Bin Packing Problem* memiliki 2 jenis masalah, pertama bagaimana mencari objek dengan ukuran tetap yang akan dimasukkan kedalam *bin*. Kedua bagaimana mencari ukuran yang tepat untuk *bin* yang akan digunakan [1].

2.1.1 Metode First Fit

Metode *First Fit* memiliki terminologi dimana semua objek atau benda P_i dimasukkan kedalam wadah secara berurutan. Dimulai dari objek P_iI sampai dengan ke objek P_in hingga wadah L_i pertama penuh lalu dilanjutkan ke wadah berikutnya. Dengan syarat semua objek P_i harus memulai kembali dimasukkan ke wadah L_i pertama jika masih muat lalu dilanjutkan ke wadah berikutnya dan tidak melebihi nilai dari wadah yang nilainya sudah ditetapkan [5][6].

```
FIRST-FIT ALGORITHM (FF)

Input: An instance a_1, \ldots, a_n of the Bin-Packing Problem.

Output: A solution (k, f).

① For i := 1 to n do:

Set f(i) := \min \left\{ j \in \mathbb{N} : \sum_{h < i: f(h) = j} a_h + a_i \le 1 \right\}.

② Set k := \max_{i \in \{1, \ldots, n\}} f(i).
```

Gambar 1 Pseudocode Metode First Fit

2.1.2 Metode First Fit Decreasing

Metode *First Fit Decreasing* hampir sama dengan metode *First Fit* biasa. Namun perbedaannya terletak pada pengurutan semua objek *Pi* dimulai dari bobot nilai yang lebih besar sampai bobot nilai yang terkecil. Untuk selanjutnya digunakan proses yang sama dengan metode First Fit [5][6].

```
FIRST-FIT-DECREASING ALGORITHM (FFD)

Input: An instance a_1, \ldots, a_n of the BIN-PACKING PROBLEM.

Output: A solution (k, f).

① Sort the numbers such that a_1 \ge a_2 \ge \ldots \ge a_n.

② Apply the FIRST-FIT ALGORITHM.
```

Gambar 2 Pseudocode Metode First Fit Decreasing

2.2 Genetic Algorithm

Genetic Algorithm atau Algoritma Genetika diperkenalkan oleh John Holland, Salah satu pendiri Computing Evolution, pada tahun 1975. Tujuannya adalah untuk membuat komputer melakukan apa yang alam lakukan. Genetic Algorithm diawali oleh urutan langkah-langkah prosedur Kromosom buatan yang bergerak dari satu populasi ke populasi baru. Menggunakan proses dan teknik yang diambil dari genetika yang dikenal sebagai seleksi alam, persilangan atau crossover dan mutasi gen. Setiap kromosom terdiri dari sejumlah gen dan setiap gen diwakili oleh nilai biner [6][7][8].

Berikut ini adalah tahapan-tahapan proses dalam Genetic Algorithm:

- 1. Input Data, data yang telah disiapkan dimasukkan sebagai inputan.
- Inisialisasi Populasi, membangkitkan populasi awal merupakan proses membangkitkan sejumlah

- individu atau kromosom secara acak atau melebihi prosedur tertentu.
- 3. Menghitung Nilai Fitness, individu yang bernilai fitness tinggi akan dijadikan acuan nilai performansinya.
- 4. Sekelsi, merupakan proses pemilihan individu yang akan dijadikan induk untuk proses selanjutnya.
- 5. Crossover, atau pindah silang merupakan proses pemilihan posisi string dan menukarnya secara acak pada induk-induk yang sudah diseleksi sebelumnya.
- 6. Mutasi, dioperasikan sebagai cara unguk mengkembalikan materi genetic yang hilang dengan cara mengganti atau menggeser nilai string yang terpilih secara acak sesuai prosedur yang ditetapkan.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) merupakan metode metaheuristic yang terinspirasi dari perilaku social binatang seperti kawanan burung dan ikan. Ketika mencari mangsa, kawanan burung tersebut bergerak bersama-sama, tanpa ada yang mengkoordinasi, tetapi pergerakannya memiliki derajat keteraturan tertentu [9].

Prosedur atau cara kerja dari PSO tersusun atas tiga bagian yaitu [9]:

- 1. Mengevaluasi nilai fitness tiap partikel.
- 2. Mengupdate nilai fitness dan posisi individu terbaik (global test)
- 3. Mengupdate kecepatan dan posisi tiap partikel.

3. Metode Penelitian

3.1 Data yang Digunakan

Data yang digunakan dalam penelitian ini adalah sederet bilangan integer sebanyak 150 data yang merepresentasikan nilai atau ukuran dari benda atau objek yang akan diproses.

Berikut ini adalah deret data yang digunakan:

```
56 53 56 49 41 51 55 51 52 48 47 45 47 53 49 48 43 46 48 48 48 53 53 48 48 46 52 53 43 48 51 53 46 44 50 53 53 57 52 51 56 50 42 44 58 48 59 43 49 53 43 50 59 49 53 47 52 47 53 44 58 50 41 48 53 48 52 46 45 52 51 46 54 55 54 56 54 56 41 37 54 51 46 49 55 43 50 57 57 43 49 50 55 41 49 46 47 54 50 46 55 59 47 47 50 53 47 57 50 49 54 53 44 48 51 55 53 54 49 47 64 49 60 56 47 48 48 50 38 45 43 47 58 45 45 72 55 49 50 49 50 51 55 53 47 49 54 53 59
```

Data diatas diambil dari data *benchmark* yang diambil sebagian (sebanyak 150 data). Dan data *benchmark* tersebut diambil dari jurnal Burk, Hyde dan Kendell [10].

Untuk ukuran wadah atau *bin* yang digunakan ditentukan sebanyak 2 ukuran wadah untuk dua kali percobaan. Pertama, ukuran wadah untuk percobaan pertama adalah 100. Dan kedua, ukuran wadah untuk percobaan kedua adalah 120.

3.2 Proses Perhitungan Lower Bound

Lower Bound atau batas indeks nilai terkecil digunakan untuk menghitung kompleksitas pada Bin Packing Problem. Untuk mengitung Lower Bound (k) digunakan rumus sebagai berikut:

$$\left| \mathbf{k} = \frac{(\mathbf{P_i} \mathbf{1} + \mathbf{P_i} \mathbf{2} + \dots + \mathbf{P_i} \mathbf{n})}{L} \right|$$

Dimana Pi adalah deret benda atau objek dengan ukuran tertentu, dan L adalah ukuran dari wadah atau bin.

Contoh diambil 10 data dari data *benchmark* yang digunakan yaitu: 56 53 56 49 41 51 55 51 52 48. Dan ukuran wadah atau bin adalah 100. Sehingga dapat dihitung:

$$k = \left| \frac{56 + 53 + 56 + 49 + 41 + 51 + 55 + 51 + 52 + 48}{100} \right|$$

$$k = |5.12|$$

$$k = 6$$

Lower Bound pada contoh diatas menunjukan nilai 5.12 yang dibulatkan menjadi 6 yaitu jumlah bin yang digunakan untuk menampung 10 objek tersebut.

3.3 Proses Metode First Fit

Pada metode $First\ Fit$, proses yang akan dikerjakan sesuai dengan pseudocode diatas dimana objek P_iI akan dimasukkan kedalam bin L_iI , jika L_iI belum penuh lihat P_i2 , apakah bisa masuk ke L_iI ? jika ya, masukkan objek kedalam L_iI , jika tidak, masukkan objek P_i2 kedalam L_i2 . Ambil objek P_i3 , cek apakah L_iI bisa masuk? Jika ya, masukkan objek P_i3 kedalam L_iI . Jika tidak, cek wadah L_i2 apakah bisa masuk? Jika ya, masukkan P_i3 kedalam P_i3 kedalam P_i3 ban begitu seterusnya.

3.4 Proses Metode First Fit Decreasing

Pada metode *First Fit Decreasing*, proses yang dikerjakan sama dengan proses metode *First Fit*. Namun data atau objek yang digunakan diurutkan dari ukuran yang paling besar ke objek dengan ukuran paling kecil. Setelah itu proses *First Fit* dilakukan.

3.5 Proses Genetic Algorithm

Untuk proses pada metode *Genetic Algorithm* parameter-parameter yang digunakan adalah:

- a. Jumlah Generasi yang digunakan adalah 1000.
- b. Jumlah Kromosom yang digunakan adalah 100.
- c. Probabilitas *Crossover* yang digunakan adalah 0.8.
- d. Probabilitas Mutasi yang digunakan adalah 0.8.

- e. Metode Seleksi yang digunakan adalah Roulette Wheel Selection.
- f. Metode *Crossover* yang digunakan adalah *Permutation Crossover*.
- g. Metode Mutasi yang digunakan adalah Swapping Mutation.

3.6 Proses Particle Swarm Optimization

Untuk proses pada metode *Particle Swarm Optimization*, parameter-parameter yang digunakan adalah:

- a. Jumlah Iterasi yang digunakan adalah 1000.
- b. Swarm Size yang digunakan adalah 50.
- c. Inertia Weight yang digunakan adalah 1.
- d. Learning Coefficient untuk Personal adalah 1.5. Dan untuk Global adalah 2.0.
- e. *Particle Speed* untuk Minimum adalah 2. Dan Maximum adalah5.
- f. Lower Bound yang digunakan adalah 0. Dan Upper Bound yang digunakan adalah 1.

4. Hasil dan Pembahasan

4.1 Pengujian Pertama

Pada pengujian pertama dimana ukuran bin yang digunakan adalah 100, hasil dari metode *Lower Bound* didapat 76 bin atau wadah, dimana nilai *Lower Bound* akan menjadi acuan jumlah bin terkecil untuk seluruh metode pada percobaan pertama.

Pengujian untuk metode *First Fit* didapat 85 bin atau wadah. Jika dibandingkan dengan Lower Bound jumlah bin yang dihasilkan lebih banyak.

Pada metode *First Fit Decreasing* didapat 80 bin atau wadah. Jumlah bin ini hampir mendekati nilai Lower Bound. Dapat dilihat metode First Fit Decreasing lebih baik dibandingkan metode First Fit.

Untuk metode *Genetic Algorithm* dilakukan pengujian sebanyak 10 kali untuk ukuran parameter yang telah ditentukan. Hasil percobaan disajikan dalam bentuk table dibawah ini.

Tabel 1 Hasil Pengujian Pertama Genetic Algorithm

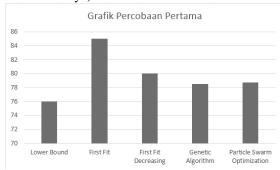
Percobaan Genetic Algorithm
76
78
77
79
80
80
79
78
80
78

Untuk metode *Particle Swarm Optimization* pengujian pertama dilakukan sebanyak 10 kali untuk masing-masing parameter. Hasil percobaan disajikan dalam bentuk table dibawah ini.

Tabel 2 Hasil Pengujian Pertama Particle Swarm
Ontimization

Optimization
Percobaan Particle Swarm
Optimization
78
78
79
77
78
80
79
79
80
79

Dari hasil percobaan pertama yang dilakukan dapat dilihat bahwa penggunaan metode *metaheuristic* seperti *Genetic Algorithm* dan *Particle Swarm Optimization* jauh lebih mendekati nilai dari *Lower Bound* dimana jika keseluruhan hasil pada percobaan pertama digambarkan menggunakan grafik seperti dibawah ini (nilai untuk grafik GA dan PSO adalah nilai rata-ratanya).



Gambar 3 Grafik Percobaan Pertama

4.2 Pengujian Kedua

Dari pengujian kedua dimana ukuran bin adalah 120, hasil pengujian untuk metode *Lower Bound* didapat 63 bin atau wadah, dimana nilai *Lower Bound* akan menjadi acuan jumlah bin terkecil untuk seluruh metode pada percobaan kedua.

Untuk hasil percobaan metode *First Fit* didapat 75 bin atau wadah. Jika dibandingkan dengan Lower Bound jumlah bin yang dihasilkan lebih banyak.

Pada metode *First Fit Decreasing* didapat 74 bin atau wadah. Jumlah bin ini hampir mendekati nilai Lower Bound. Dapat dilihat metode First Fit Decreasing lebih baik dibandingkan metode First Fit.

Untuk metode *Genetic Algorithm* dilakukan pengujian sebanyak 10 kali untuk ukuran parameter

yang sudah ditentukan. Hasil percobaan disajikan dalam bentuk table dibawah ini.

Tabel 3	Hasil	Pengujian	Kedua	Genetic	Algorithm

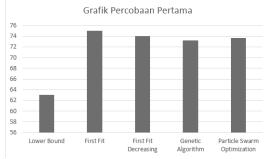
Percobaan Genetic Algorithm
74
74
74
73
70
74
73
74
74
72

Untuk metode Particle Swarm Optimization pengujian pertama dilakukan sebanyak 10 kali untuk masing-masing parameter. Hasil percobaan disajikan dalam bentuk table dibawah ini.

Tabel 4 Hasil Pengujian Kedua Particle Swarm

Optimization
Percobaan Particle Swarm
Optimization
73
74
74
75
74
73
74
74
73
73

Dari hasil percobaan kedua yang dilakukan dapat dilihat bahwa penggunaan metode metaheuristic seperti Genetic Algorithm dan Particle Swarm Optimization masih jauh lebih mendekati nilai dari Lower Bound namun hasil dari First Fit dan First Fit Decreasing tidak terlalu jauh perbedaannya dan jika keseluruhan hasil pada percobaan pertama digambarkan menggunakan grafik seperti dibawah ini.



Gambar 4 Grafik Percobaan Kedua

5. Kesimpulan

Dari hasil analisis, pengujian serta pembahasan diatas dapat disimpulkan bahwa:

- 1. Dari hasil pengujian yang dilakukan metode metaheuristic lebih optimal dibandingkan dengan metode First Fit dan First Fit Decreasing.
- 2. Pada percobaan pertama bin atau wadah yang dihasilkan jauh lebih untuk metode metaheuristic yaitu Genetic Algorithm dan Particle Swarm Optimization jauh lebih sedikit dan mendekati hasil dari Lower Bound.
- 3. Pada percobaan kedua bin atau wadah yang dihasilkan tidak terlalu jauh berbeda antara Genetic Algorithm dan Particle Swarm Optimization dengan First Fit dan First Fit Decreasing.
- 4. Walaupun hasil yang diberikan jauh lebih optimal untuk metode metaheuristic (GA dan PSO) dibandingkan metode First Fit dan First Fit Decreasing, tetapi masih belum bisa mereduksi atau melampaui hasil dari Lower Bound.

6. References

- [1] Martello, S., and Toth, P., Knapsack Problem: Algorithms and Computer Implementations, John Wiley & Sons Ltd, England, 1990.
- [2] Panggabean, T.N., "Analisis Tingkat Optimasi Algoritma Genetika Dalam Hukum Ketetapan Hardy-Weinberg pada Bin Packing Problem", Journal Of Computer Engineering Vol. 1 No. 2, System, and Science, Universitas Negeri Medan, Medan, 2016, pp. 12-18.
- [3] Sivanandam, S. N., and Deepa, S. N., Introduction to Genetic Algorithms, Springer, Berlin, 2008.
- [4] Weise, T., Global Optimization Algorithms: Theory and Application Second Edition, Self-Published, Online available at http://www.it-weise.de/, 2009.
- [5] Harren, R., Two-Dimensional Packing Problem, Dissertation, Universitat des Saarlandes, 2010, pp. 1-131
- [6] Panggabean, T.N., Analisis Reduksi NP-Hard Bin Packing Problem dalam Algoritma Genetika pada Hukum Ketetapan Hardy-Weinberg, Thesis, Universitas Sumatera Utara, Medan, 2016.
- [7] Negnevitsky, M., Artificial Intelligence: A Guide to Intelligent Systems, Second Edition, Addison-Wesley, Harlow, 2005.
- [8] Perdana, A., Analisis Performansi pada Penerapan Hukum Ketetapan Hardy-Weinberg dalam Algoritma Genetika, Thesis, Universitas Sumatera Utara, Medan, 2014

- [9] Purnomo, H.D., Cara Mudah Belajar Metode Optimisasi Metaheuristik Menggunakan Matlab, Gava Media, Yogyakarta, 2014.
- [10] Burke, E.K., Hyde, M.R., and Kendall, G., Providing a Memory Mechanism to Enchance the Evolutionary Design of

Heuristics, Proceedings of the IEEE World Congress on Computational Intelligence, Barcelona, 2010, pp. 3884-3890.