

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : Muhammad Fahry Aritianta
NRP : 5110100012
DOSEN WALI : Daniel O. Siahaan, S.Kom., M.Sc., PDEng
DOSEN PEMBIMBING : 1. Arya Yudhi Wijaya, S.Kom., M.Kom.
2. Rully Soelaiman, S.Kom., M.Kom.

2. JUDUL TUGAS AKHIR

“Identifikasi Parameter Optimal Metode Meta-heuristik *Particle Swarm Optimization* dan *Tabu Search* pada Penyelesaian Penjadwalan *Flow Shop*”

3. LATAR BELAKANG

Seiring dengan kemajuan teknologi, semakin banyak mesin yang digunakan untuk mengerjakan sebuah tugas yang sebelumnya dilakukan oleh manusia. Dalam berbagai bidang terutamanya *manufacturing*, banyak mesin yang digunakan untuk menyelesaikan suatu pekerjaan. Penjadwalan tugas multi-mesin merupakan permasalahan yang muncul dari penggunaan mesin yang banyak di mana suatu tugas dikerjakan oleh lebih dari satu mesin.

Masalah utama ketika memberikan pekerjaan kepada mesin-mesin ini adalah kurang optimalnya penggunaan mesin untuk menyelesaikan sebuah urutan kerja. Optimalnya sebuah mesin dapat dilihat dari waktu kerja yang dihasilkan untuk menyelesaikan suatu urutan masalah. Semakin kecil waktu yang dibutuhkan maka semakin optimal mesin tersebut dan begitu juga sebaliknya. Pembuatan jadwal dibutuhkan agar mesin-mesin tersebut dapat menyelesaikan sebuah urutan kerja dengan waktu kerja seminimal mungkin. Bila jadwal kerja tidak diberikan secara benar kepada mesin, maka akan banyak mesin yang menganggur sehingga pekerjaan

akan terselesaikan dalam waktu lama. Permasalahan penjadwalan dengan sebuah urutan kerja disebut dengan penjadwalan *flow shop*. Permasalahan penjadwalan seperti ini sangat sulit dilakukan walau dengan bentuk paling sederhana sekalipun.

Tujuan dari permasalahan ini adalah untuk memaksimalkan kinerja mesin dengan meminimalkan waktu kerja yang dilakukan oleh mesin dalam menyelesaikan suatu urutan kerja. Permasalahan penjadwalan *flow shop* adalah permasalahan penjadwalan dengan m mesin dan n pekerjaan.

Penjadwalan *flow shop* untuk meminimalisasi waktu kerja merupakan *NP-Hard* [1]. Salah satu cara untuk menyelesaikan permasalahan penjadwalan *flow shop* adalah dengan menggunakan metode heuristik. Penelitian yang dilakukan oleh Xia dan Wu mengajukan sebuah metode untuk menyelesaikan penjadwalan *job shop* dengan menggunakan metode *Particle Swarm Optimization* (PSO) dan *Simulated Annealing* (SA) [2]. Solusi yang dihasilkan dari penelitian ini memiliki kualitas yang hampir sama dengan solusi yang dihasilkan oleh metode lainnya seperti *Genetic Algorithm* dan *Ant Colony System*. Penemuan ini kemudian memotivasi F. Choong, S. Phon-Amnuaisuk, dan M. Y. Alias untuk menggabungkan PSO dengan metode meta-heuristik lainnya yaitu *Tabu Search* (TS) dan mencobanya pada permasalahan penjadwalan *flow shop* [1].

Permasalahan penjadwalan yang diangkat pada Tugas Akhir ini adalah penjadwalan *flow shop*. Pada Tugas Akhir ini akan dilakukan implementasi metode yang diajukan oleh F. Choong, S. Phon-Amnuaisuk, dan M. Y. Alias yang berupa metode PSO-TS. Metode ini dipilih karena PSO dapat memberikan hasil yang memiliki kualitas yang sama dengan metode heuristik lainnya namun dengan implementasi yang lebih mudah [1]. Metode TS dipilih untuk meningkatkan solusi yang dihasilkan oleh PSO dan menghindari pencarian solusi yang dilakukan PSO untuk terjebak pada nilai *local minima* [3].

Hasil yang diharapkan dari Tugas Akhir ini adalah metode PSO-TS dapat menemukan waktu kerja yang optimal pada permasalahan penjadwalan *flow shop*.

4. RUMUSAN MASALAH

Tugas Akhir ini mengangkat beberapa rumusan masalah, yaitu sebagai berikut:

1. Penerapan metode meta-heuristik PSO-TS untuk mendapatkan solusi dari penjadwalan mesin yang optimal.
2. Perancangan desain algoritma PSO-TS untuk menyelesaikan penjadwalan *flow shop*.
3. Melakukan implementasi metode PSO-TS untuk menyelesaikan penjadwalan *flow shop*.

5. BATASAN MASALAH

Tugas Akhir ini mempunyai banyak permasalahan. Tetapi tidak semua permasalahan akan terselesaikan karena keterbatasan sumber daya dari penulis. Maka dibuat batasan-batasan permasalahan yang ada, yaitu sebagai berikut:

1. Semua mesin terus tersedia sejak waktu 0.
2. Tiap mesin tidak dapat menangani lebih dari satu tugas pada satu waktu.
3. Waktu pengolahan dan jumlah mesin yang dibutuhkan pada tiap tahapnya diketahui diawal.
4. Waktu pengaturan dan waktu komunikasi antar mesin terdapat pada waktu proses dan bersifat independen dari urutan kerja.
5. Algoritma yang digunakan adalah *Particle Swarm Optimization* dengan algoritma meta-heuristik *Tabu Search*.
6. Implementasi algoritma menggunakan bahasa C++ dengan *software* Dev-C++.
7. *Dataset* yang digunakan adalah *dataset* dengan mesin *random* dan dengan mesin yang telah ditentukan. Pada *dataset* mesin *random*, jumlah mesin pada tiap tahapnya diberikan secara acak. Sedangkan pada *dataset* dengan mesin ditentukan, jumlah mesin ditentukan pada tiap tahapnya sebelum diproses oleh *software*.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Mempelajari cara penggabungan metode PSO dengan metode TS menjadi metode meta-heuristik PSO-TS dan cara kerja dari PSO-TS.
2. Memahami desain metode meta-heuristik PSO-TS untuk menyelesaikan penjadwalan *flow shop*.
3. Melakukan implementasi metode meta-heuristik PSO-TS untuk menyelesaikan penjadwalan *flow shop*.
4. Mengevaluasi kinerja dari metode meta-heuristik PSO-TS untuk menyelesaikan penjadwalan *flow shop* dengan melakukan uji coba.

7. MANFAAT TUGAS AKHIR

Tugas Akhir ini diharapkan dapat membantu meminimalkan waktu kerja dari satu atau lebih mesin yang digunakan untuk menyelesaikan suatu tahapan kerja pada studi kasus *manufacturing*. Minimalisasi waktu kerja dari satu atau lebih mesin dilakukan dengan cara menerapkan metode PSO-TS.

8. TINJAUAN PUSTAKA

Penulis menggunakan beberapa tinjauan pustaka, yaitu sebagai berikut:

1. Penjadwalan *Flow Shop*

Penjadwalan *flow shop* merupakan perluasan dari penjadwalan mesin dan merupakan bagian dari penjadwalan *multiprocessor*. Pada penjadwalan *flow shop* terdapat m mesin yang harus memproses n pekerjaan. Semua pekerjaan memiliki urutan proses yang sama melalui mesin. Urutan pekerjaan yang

dilakukan pada setiap mesin dapat berbeda satu dengan yang lainnya. Seiring dengan kemajuan teknologi yang terjadi, banyak permasalahan pada dunia nyata yang menggunakan mesin paralel seperti *manufacturing*, simulasi *power system*, dan pengendali lalu lintas pada daerah terbatas [1]. Permasalahan dari penjadwalan *flow shop* adalah permasalahan *NP-Hard*. Kompleksitas dari permasalahan ini bisa bertambah dengan bertambahnya jumlah tahapan yang harus dilalui. Solusi dari permasalahan ini adalah nilai waktu kerja yang seminimal mungkin. Maka dari itu untuk mendapatkan nilai solusi dari permasalahan *flow shop*, digunakan algoritma meta-heuristik PSO-TS.

2. Particle Swarm Optimization

Particle Swarm Optimization diperkenalkan oleh James Kennedy dan Russell Eberhart pada tahun 1995 [4]. Algoritma ini menggunakan beberapa partikel yang merupakan sebuah kawanan yang bergerak pada dalam ruang pencarian untuk menemukan solusi terbaik. Setiap partikel yang ada pada ruang pencarian diperlakukan sebagai sebuah titik yang menyesuaikan kecepatan gerakannya berdasarkan dari pengalaman gerak partikel itu sendiri dan partikel lainnya. Pada algoritma ini, dikenal *pBest* dan *gBest*. *pBest* adalah nilai dari solusi terbaik yang didapatkan oleh sebuah partikel pada ruang pencarian setelah melakukan penelusuran jejak untuk koordinat partikel tersebut. Sedangkan *gBest* adalah nilai dari solusi terbaik dari sebuah partikel berdasarkan dari jejak kumpulan partikel tersebut [5].

Persamaan 1 dan 2 digunakan oleh algoritma PSO dasar untuk mengetahui posisi dan kecepatan sebuah partikel [1].

$$V_{id} = WV_{id} + C_1 R_1 (P_{id} - X_{id}) + C_2 R_2 (P_{gd} - X_{id}) \quad (1)$$

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

Persamaan 1 adalah untuk mencari kecepatan dari sebuah partikel pada PSO. Sedangkan Persamaan 2 adalah untuk mencari posisi baru dari sebuah partikel. Dengan W adalah berat inersia. X_{id} adalah posisi dari partikel. P_{id} adalah solusi lokal terbaik atau disebut juga *pBest*. P_{gd} adalah solusi global terbaik atau disebut juga *gBest*. C_1 dan C_2 adalah akselerasi konstan dari sebuah partikel. R_1 dan R_2 adalah nilai *random* dengan rentang 0 sampai 1.

Sedangkan untuk pengerjaan Tugas Akhir ini, digunakan Persamaan 3 untuk menemukan fungsi eksponensial untuk menentukan *inertia weight* [1].

$$W = W_{end} + (W_{start} - W_{end}) e^{\frac{x \times x}{x_{max}}} \quad (3)$$

3. Tabu Search

Tabu Search diperkenalkan oleh Fred W. Glover pada tahun 1986. Konsep dasar dari *Tabu Search* adalah sebuah algoritma meta-heuristik yang ditumpangkan pada algoritma heuristik lainnya, yang dalam kasus ini adalah

Particle Swarm Optimization. Pendekatan dasar dari *Tabu Search* adalah dengan melarang bergerak sebuah partikel dengan membawa nilai solusi sementara pada iterasi berikutnya, ke titik pada ruang solusi yang sebelumnya dikunjungi. Daerah yang telah dikunjungi dapat dilacak dengan menggunakan memori jangka pendek lalu TS melarang sebuah partikel untuk bergerak kembali ke posisi yang telah dikunjungi sebelumnya. Lingkungan solusi pada saat seperti itu terbatas pada solusi yang terdapat pada *tabu list* yang selanjutnya kita sebut sebagai *allowed list*. Pada setiap iterasi solusi terbaik dari *allowed set* dipilih menjadi solusi sementara yang baru. Setelah itu solusi ini ditambahkan ke dalam *tabu list* dan salah satu solusi yang telah ada di dalam *tabu list* dihapus, biasanya dalam urutan FIFO (*first in first out*). Metode TS akan berhenti ketika kondisi terminasi ditemukan atau apabila *allowed set* kosong di mana pada kondisi ini semua solusi yang tersedia telah dilarang oleh *tabu list*. Dengan hal ini, *Tabu Search* menjamin untuk menemukan sebuah daerah baru pada ruang pencarian dan akan menghindari dari situasi terjebak pada nilai *local minima* yang dapat dihasilkan oleh *Particle Swarm Optimization*, dan pada akhirnya akan mendapatkan solusi yang diinginkan. *Tabu Search* terdiri dari beberapa elemen yaitu *specification of a neighborhood structure*, *the move attributes*, *the tabu list length*, *aspiration criterion*, dan *stopping rules* [3]. Algoritma TS dapat dilihat pada Gambar 1.

```

s ← GenerateInitialSolution()
InitializeTabuLists(TL1, ..., TLr)
k ← 0
while termination conditions not met do
    AllowedSet(s,k) ← {s' ∈ N(s) | s does not violate a tabu
                                condition, or it satisfies at least
                                one aspiration condition}
    s ← ChooseBestOf(AllowedSet(s,k))
    UpdateTabuListsAndAspirationConditions()
    k ← k + 1
endwhile

```

Gambar 1. Algoritma *Tabu Search*

4. Metode Meta-heuristik

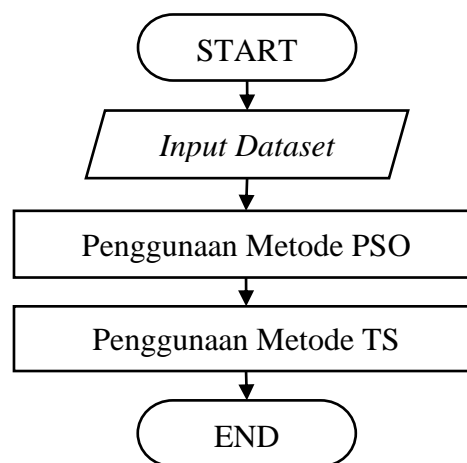
Meta-heuristik adalah strategi tingkat tinggi untuk menjelajahi ruang pencarian dengan metode yang berbeda. Metode ini memecahkan permasalahan dengan melakukan perbaikan mulai dengan satu atau lebih solusi awal. Solusi awal ini dapat dihasilkan secara acak dan dapat pula dihasilkan berdasarkan heuristik tertentu. Metode-metode meta-heuristik menggabungkan *neighborhood search*, *memory structure*, dan *recombination* pada tiap solusi untuk menghasilkan solusi yang lebih baik.

Setidaknya ada tujuh metode meta-heuristik untuk menyelesaikan permasalahan kombinatorial termasuk penjadwalan. Ketujuh metode meta-

heuristik tersebut terbagi atas dua kategori yaitu perluasan dari *Single Solutions Metaheuristics* dan *Population Metaheuristics* [6]. Metode perluasan dari *Single Solutions Metaheuristics* bertujuan untuk menghindari terjadinya *local minima* sehingga dapat melakukan eksplorasi pada ruang pencarian sambil terus mencari nilai *local minima* yang lebih baik. Sedangkan metode *Population Metaheuristics* bertujuan untuk mencoba mengidentifikasi area yang berkualitas pada ruang pencarian dengan cara melakukan *sampling*.

9. RINGKASAN ISI TUGAS AKHIR

Tugas Akhir ini akan melakukan implementasi terhadap metode *Particle Swarm Optimization* dengan *Tabu Search*. Secara garis besar proses implementasi ini melalui tahapan-tahapan yang dijelaskan pada diagram alur pada Gambar 2.



Gambar 2. Diagram Alur untuk Metode PSO-TS

Berikut penjelasan beberapa tahapan implementasi metode PSO-TS yaitu:

1. *Input Dataset*

Pada tahapan ini *dataset* yang akan digunakan dimasukkan ke dalam program yang telah dibangun berdasarkan metode PSO-TS. *Dataset* yang dimasukkan berupa jumlah mesin yang akan digunakan. *Dataset* ini bisa berupa nilai acak dan bisa pula berupa masukan yang diberikan.

2. Penggunaan Metode PSO

Pada tahapan ini *dataset* yang telah dimasukkan sebelumnya diproses dengan metode PSO. Metode PSO akan menghasilkan nilai *pBest* dan *gBest*. Jika nilai *pBest* yang baru lebih baik dari nilai yang lama, maka nilai *pBest* akan diperbarui dan posisi dari *pBest* sebelumnya juga diperbarui berdasarkan posisi *pBest* yang baru. Nilai *gBest* juga akan berubah jika nilai *gBest* yang baru lebih baik dari yang lama. Bila kondisi terminasi telah

terpenuhi, nilai *gBest* yang terakhir akan digunakan untuk inisialisasi metode TS.

3. Penggunaan Metode TS

Pada tahapan ini metode TS digunakan untuk mencegah kondisi terjebak pada nilai *local minima* dan memberikan daerah baru untuk dijelajahi pada ruang pencarian solusi. Pencarian yang dilakukan oleh metode TS berawal dari nilai *gBest* yang dihasilkan oleh metode PSO. Nilai ini akan dimasukkan ke dalam *tabu list*. Metode TS kemudian akan menghasilkan sekumpulan *sub-neighborhoods* berdasarkan *interchange neighborhood* di mana dua pekerjaan yang ada pada daftar pekerjaan ditukar. Sekumpulan *neighborhoods* itu kemudian dilakukan evaluasi untuk menemukan solusi baru selain yang ada pada *tabu list*. Jika nilai solusi yang lebih baik ditemukan, maka *tabu list* akan diperbarui biasanya dengan aturan FIFO (*first in first out*).

10. METODOLOGI

a. Penyusunan proposal tugas akhir

Tahap awal pengerjaan Tugas Akhir adalah menyusun proposal Tugas Akhir. Pada proposal Tugas Akhir ini, diajukan gagasan untuk membantu menyelesaikan masalah yang terjadi pada penjadwalan *flow shop* dengan menggunakan algoritma meta-heuristik PSO-TS.

b. Studi literatur

Tahap berikutnya adalah mencari informasi dan studi literatur apa saja yang bisa dijadikan referensi untuk melakukan pengerjaan Tugas Akhir. Informasi dan studi literatur ini didapatkan dari buku, *internet*, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

c. Implementasi perangkat lunak

Implementasi merupakan tahap untuk membangun algoritma tersebut. Untuk membangun algoritma yang telah dirancang sebelumnya, implementasi dibangun dengan menggunakan Dev-C++.

d. Pengujian dan evaluasi

Tahap berikutnya adalah melakukan uji coba dari metode dan model menggunakan *dataset* dengan mesin *random* dan dengan mesin yang telah ditentukan. Pada *dataset* dengan mesin *random* jumlah mesin pada tiap tahapnya diberikan secara acak. Sedangkan pada *dataset* dengan mesin ditentukan, jumlah mesin ditentukan pada tiap tahapnya sebelum diproses oleh *software*. Pada tahap ini juga dilakukan evaluasi kekurangan-kekurangan apa

saja yang didapat dari uji coba tersebut. Diharapkan dengan melakukan evaluasi tersebut maka program dapat disempurnakan.

e. Penyusunan Buku Tugas Akhir

Dalam tahap akhir ini, penulis melakukan penyusunan laporan yang berisikan dokumen pembuatan dan hasil pengerjaan pada perangkat lunak yang telah dibuat. Secara garis besar, sistematika penulisan buku Tugas Akhir adalah sebagai berikut:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Jadwal kegiatan dapat dilihat pada Tabel 1.

Tabel 1. Jadwal kegiatan

Tahapan	2014																			
	Februari					Maret				April				Mei				Juni		
Penyusunan proposal																				
Studi literatur																				
Implementasi																				
Pengujian dan evaluasi																				
Penyusunan buku																				

12. DAFTAR PUSTAKA

- [1] F. Choong, S. Phon-Amnuaisuk and M. Alias, "Metaheuristic methods in hybrid flow shop scheduling problem," *Expert Systems with Applications*, p. 10787–10793, 2011.

- [2] W. Xia and Z. Wu, "A Hybrid Particle Swarm Optimization Approach for the Job-Shop Scheduling," *International Journal of Advance Manufacturing Technology*, no. 4222, pp. 146-155, 2006.
- [3] G. Zhang, X. Shao, P. Li and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," *Computers & Industrial Engineering*, no. 59, p. 1309–1318, 2009.
- [4] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Washington, DC, 1995.
- [5] D. Sha and C.-Y. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Computers & Industrial Engineering*, no. 51, pp. 791-808, 2006.
- [6] M. Gendreau and J.-Y. Potvin, "Metaheuristics in Combinatorial Optimization," *Annals of Operations Research*, no. 140, pp. 189-213, 2005.