

PENGEMBANGAN ALGORITMA *DIFFERENTIAL EVOLUTION* UNTUK PENJADWALAN *FLOW SHOP* MULTI OBYEKTIF DENGAN BANYAK MESIN

Rudi Nurdiansyah

Jurusan Teknik Industri, Fakultas Teknologi Industri,

Institut Teknologi Sepuluh Nopember Surabaya

rudy_nurdian@yahoo.co.id

ABSTRAK

Paper ini akan menyajikan permasalahan penjadwalan *flow shop* dengan mempertimbangkan 2 obyektif, yaitu *makespan* dan *total flowtime*. Beberapa algoritma telah dikembangkan untuk menyelesaikan permasalahan tersebut. Paper ini menyajikan algoritma *Differential Evolution* yang ditambah dengan *adaptive parameters* serta strategi *local search* untuk menyelesaikan permasalahan penjadwalan tersebut. Algoritma yang diusulkan pada riset ini akan diuji dengan problem-problem yang ada dalam literatur. Performa dari algoritma yang diusulkan dibandingkan dengan algoritma-algoritma yang dikembangkan oleh Rajendran (1995), Ravindran *et al* (2005) serta Yagmahan dan Yenisey (2010). Penelitian ini menunjukkan bahwa metode DE yang dikembangkan dalam penelitian ini lebih efisien dan mempunyai performa yang lebih baik dibandingkan dengan metode-metode lain.

Kata Kunci: Penjadwalan *flowshop*, multi obyektif, *Makespan*, *Total flowtime*, *Differential Evolution*

ABSTRACT

This paper considers the flow shop scheduling problem with respect to the both objectives of makespan and total flowtime. Several algorithms have been proposed to solve this problem. This paper presents Differential Evolution algorithm which improve with adaptive parameters and local search strategy in order to solve this scheduling problem. The proposed algorithm is tested with well-known problems in literature. The performance of proposed algorithm was compared with algorithms developed by Rajendran (1995), Ravindran (2005) and Yagmahan & Yenisey (2010). The computational results show that proposed algorithm is more efficient and better than other methods compared.

Keywords: Flow shop scheduling, Multi-objective, Makespan, Total flowtime, Differential Evolution

1. PENDAHULUAN

Permasalahan penjadwalan *flow shop* menjadi problem optimasi kombinatorial seiring dengan bertambahnya jumlah *job* dan jumlah mesin (Taillard, 1993). Problem optimasi kombinatorial merupakan *NP-hard* dan pendekatan yang lebih menjadi pilihan dari permasalahan ini adalah teknik solusi yang mendekati optimal (Yagmahan dan Yenisey, 2008). Dalam beberapa tahun terakhir, pendekatan metaheuristik seperti

Simulated Annealing (SA), *Tabu Search (TS)*, *Ant Colony Optimization (ACO)*, *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, *Differential Evolution (DE)* dan *Artificial Immune Systems (AIS)* banyak digunakan untuk memecahkan permasalahan optimasi kombinatorial karena terbukti memiliki kinerja komputasi yang baik (Yagmahan dan Yenisey, 2008).

Kebanyakan studi mengenai penjadwalan *flow shop* fokus pada minimasi *makespan*. Pada kenyataannya, banyak tujuan lain selain *makespan* yang bisa dipertimbangkan sebagai obyektif seperti *total flowtime* yang juga merupakan ukuran kinerja yang sangat penting dalam meminimalkan ongkos penjadwalan total (Yagmahan dan Yenisey, 2010). Oleh karena itu, dalam riset ini akan mempertimbangkan 2 obyektif yaitu minimasi *makespan* dan *total flowtime*. Minimasi *makespan* mengarah ke utilisasi dalam menjalankan produksi, sedangkan minimasi *flowtime* menghasilkan konsumsi yang stabil terhadap sumber daya, perputaran job yang cepat serta meminimalkan *work in process inventory* (Yagmahan dan Yenisey, 2010).

Salah satu algoritma yang mempunyai reputasi sebagai metode optimasi global optima yang efektif adalah *Differential Evolution (DE)*. Keunggulan dari DE adalah konsep yang sederhana, implementasi yang mudah dan cepat konvergen, namun performa DE sangat tergantung dari parameternya (Qian *et al* 2008). Tvrdik (2006) menyebutkan bahwa efisiensi pencarian dari DE sangat sensitif terhadap penentuan nilai parameter F dan Cr . Dalam riset ini akan dilakukan pengontrolan nilai parameter F dan Cr yang disebut *adaptive parameters*. Prosedur dari *adaptive parameters* adalah dengan menghitung nilai parameter F dan Cr pada tiap generasi dengan formula tertentu sehingga nilai parameter F dan Cr tiap generasi berubah-ubah. Algoritma DE pada penelitian ini juga akan digabungkan dengan teknik-teknik *local search* untuk meningkatkan performa dari DE. Pan *et al* (2008) menyebutkan bahwa *local search* dapat memingkatkan performa dari DE. Oleh karena itu, riset ini akan mengembangkan algoritma *Differential Evolution* dengan melakukan *adaptive parameters* terhadap parameter DE dan menambahkan prosedur *local search* sebagai tahap *improvement* dari DE. Performa dari algoritma yang dikembangkan dalam penelitian ini akan dibandingkan dengan algoritma CR(MC) yang dikembangkan oleh Rajendran (1995), HAMC oleh Ravindran *et al* (2005) serta MOACSA oleh Yagmahan dan Yenisey (2010).

2. TINJAUAN PUSTAKA

Deskripsi model penjadwalan *flow shop* menurut Hejazi dan Saghafian (2005) adalah sebagai berikut : Seperangkat M mesin $M=\{1,2,\dots,m\}$ yang digunakan untuk memproses sekelompok N jobs, $N=\{1,2,\dots,n\}$. Pada satu titik waktu, setiap mesin hanya dapat memproses satu *job* dan setiap *job* hanya diproses di satu mesin. Setiap *job* di tahap operasi i , hanya diproses sekali saja di salah satu mesin. Keseluruhan *job* dikerjakan dalam arah aliran operasi yang sama. *Mix integer programming (MIP)* dari pemasalahan penjadwalan *flow shop* adalah sebagai berikut (Seda, 2007) :

Simbol yang digunakan :

- J : Jumlah *job* (1,2,...,n)
- M : Jumlah mesin (1,2,...,m)
- O : Jumlah operasi (1,2,...,m)
- J_i : *Job* ke- i
- π_i : *Job* yang ditempatkan pada urutan ke- i
- P_{ik} : Waktu proses *job* ke- i pada mesin k

v_{ik} : *Idle time* mesin k sebelum memproses *job* ke-i
 w_{ik} : *Waiting time job* ke-i setelah dikerjakan pada mesin k sebelum dikerjakan pada mesin k+1

Variabel keputusan yang digunakan :

$$x_{ij} = \begin{cases} 1, & \text{jika job } j \text{ ditempatkan pada urutan ke-} i \\ 0, & \text{jika job } j \text{ tidak ditempatkan pada urutan ke-} i \end{cases}$$

Model Permasalahan

Minimize C_{\max} dan total F

Subject to

$$\forall i \in J: \sum_{j=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (1)$$

$$\forall j \in J: \sum_{i=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad (2)$$

$(\forall i \in J - \{n\})(\forall k \in M - \{m\})$:

$$v_{i+1,k} + \sum_{j=1}^n P_{jk} x_{i+1,j} + w_{i+1,k} = w_{ik} + \sum_{j=1}^n P_{j,k+1} x_{ij} + v_{i+1,k+1} \quad (3)$$

$$\forall k \in M - \{1\}: v_{1k} = \sum_{r=1}^{k-1} \sum_{j=1}^n P_{jr} x_{1j} \quad (4)$$

$$\forall k \in M - \{m\}: w_{1k} = 0 \quad (5)$$

$$C_{\max} = \sum_{i=1}^n (v_{in} + \sum_{j=1}^n P_{jn} x_{ij}) \quad (6)$$

Kendala suatu *job* hanya dapat dijadwalkan sekali dan satu urutan jadwal hanya terdapat satu *job* yang sama (1),(2). Kendala yang menunjukkan syarat *job* yang berurutan pada setiap mesin (3). Kendala *idle time* setiap mesin sebelum dimulainya *job* urutan pertama (4). Kendala *waiting time job* urutan pertama pada semua mesin (5). Kendala untuk menghitung total waktu penyelesaian keseluruhan *job* (6).

- Untuk menghitung *makespan* :
 $C_{\max} = C(\pi_n, m)$
- Untuk menghitung *total flow time* :

$$F = \sum_{i=1}^n C(\pi_i, m)$$

Multi Obyektif = $w_1 \cdot C_{\max} + w_2 \cdot F$

Dimana : w_1 = bobot *makespan*

w_2 = bobot *total flowtime*

3. METODOLOGI PENELITIAN

Data yang digunakan pada riset ini terdapat pada *OR-Library* (url: <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>) yaitu data kasus penjadwalan *flow shop* yang berukuran 20 *job* 5 mesin (ta001-010), 20 *job* 10 mesin (ta011-020) dan 20

job 20 mesin (ta021-028). Langkah selanjutnya adalah mengembangkan algoritma untuk menyelesaikan permasalahan penjadwalan *flow shop*. Algoritma yang digunakan dalam riset ini adalah *Differential evolution* (DE). *Differential evolution* (DE) merupakan salah satu metoda metaheuristik terbaru yang diperkenalkan oleh Storn dan Price (1995). Riset ini akan mengembangkan algoritma *Differential Evolution* dengan melakukan *adaptive parameters* terhadap parameter DE dan menambahkan prosedur *local search* sebagai tahap *improvement* dari DE. Langkah-langkah pengembangan algoritma yang dilakukan adalah sebagai berikut:

1. Inisialisasi Populasi.

Sebelum melakukan inisialisasi terhadap titik populasi maka perlu dilakukan penentuan batas atas (*ub*) dan batas bawah (*lb*). Untuk pembangkitan nilai awal generasi $g = 0$, variable ke-*j* dan *vector* ke-*i* bisa diwakili dengan notasi berikut :

$$x_{j,i,0} = lb_j + rand_j(0, 1)(ub_j - lb_j)$$

Bilangan *random* dibangkitkan dengan fungsi *rand*, dimana bilangan yang dihasilkan terletak antara (0, 1).

2. Mutasi.

Setelah tahapan inisialisasi, DE akan memutasi dan mengkombinasi populasi awal untuk menghasilkan populasi dengan ukuran N vektor percobaan. Dalam DE, mutasi dilakukan dengan cara menambahkan perbedaan dua vektor terhadap vektor ketiga dengan secara acak. Formulasinya sebagai berikut:

$$v_{i,g} = xr_{0,g} + F(xr_{1,g} - xr_{2,g})$$

Faktor skala, $F \in (0, 1)$ adalah bilangan *real* positif yang mengendalikan tingkat pertumbuhan populasi. Pada langkah ini nilai parameter *F* tiap generasi akan berubah-ubah dengan menghitung nilai parameter pada tiap generasi dengan formula yang dikembangkan oleh Tvrđik (2006) sebagai berikut :

$$F = \begin{cases} \max(F_{\min}, 1 - |\frac{f_{\max}}{f_{\min}}|) & \text{if } |\frac{f_{\max}}{f_{\min}}| < 1 \\ \max(F_{\min}, 1 - |\frac{f_{\min}}{f_{\max}}|) & \text{otherwise,} \end{cases}$$

Dimana *fmin* adalah nilai fungsi minimum dari populasi dan *fmax* adalah nilai fungsi maksimum dari populasi. *Fmin* merupakan *input parameter* yang memastikan $F \in [F_{\min}, 1]$. Formulasi ini mencerminkan pencarian yang lebih beragam pada tahap awal dan lebih intensif pada tahap berikutnya (Tvrđik, 2006).

3. *Crossover*

Pada tahap ini DE menyilangkan setiap vektor $x_{i,g}$, dengan vektor mutan $v_{i,g}$, untuk membentuk vektor hasil persilangan $u_{i,g}$.

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } (rand_j(0,1) \leq Cr, \quad \text{or} \quad j = j_{rand}) \\ x_{j,i,g} & \text{sebaliknya} \end{cases}$$

Probabilitas *crossover*, $Cr \in (0,1)$ adalah nilai yang didefinisikan untuk mengendalikan fraksi nilai parameter yang disalin dari mutan. Probabilitas *crossover* untuk tiap generasi akan ditentukan dengan rumus Mingyong dan Erbao (2010):

$$Cr = Cr_{min} + G \frac{Cr_{max} - Cr_{min}}{MAXGEN}$$

Cr adalah *crossover probability*, Cr_{min} adalah nilai terkecil dari *crossover probability* sedangkan Cr_{max} adalah nilai maksimum dari *crossover probability*. G adalah iterasi pada saat waktu t *running time* sedangkan $MAXGEN$ adalah jumlah maksimum iterasi yang diujicobakan. Tujuan dari penentuan nilai Cr dalam penelitian Mingyong dan Erbao (2010) ini adalah meningkatkan keragaman *vector* yang akan mengalami *crossover* dan menghindari dari *local optima*.

4. Selection

Jika *trial vector* $u_{i,g}$, mempunyai fungsi tujuan lebih kecil dari fungsi tujuan vektor targetnya yaitu $x_{i,g}$, maka $u_{i,g}$ akan menggantikan posisi $x_{i,g}$ dalam populasi pada generasi berikutnya. Jika sebaliknya, target akan tetap pada posisinya dalam populasi.

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } (f(u_{i,g}) \leq f(x_{i,g})) \\ x_{i,g} & \text{sebaliknya} \end{cases}$$

5. Local Search

Pada tahap ini, hasil dari seleksi akan dikenai prosedur *local search*. Prosedur *local search* yang digunakan adalah *insert-based local search*. *Insert-based local search* cenderung mengarahkan pencarian ke daerah solusi yang menjanjikan dalam waktu relatif singkat (Qian *et al*, 2008). Prosedur dari *insert-based local search* adalah dengan menyisipkan *job u* ke posisi *job v* (dimana $u \neq v$). Proses mutasi, *crossover*, seleksi dan *local search* akan diulang sampai *stopping criterion* tertentu dicapai.

6. Kriteria Pemberhentian

Dalam riset ini kriteria pemberhentian yang digunakan adalah iterasi maksimal.

Selanjutnya akan dibuat kode program algoritma pada *software* Matlab. Pengujian algoritma dilakukan dengan menjalankan kode yang telah dibuat dengan menggunakan spesifikasi komputer Intel Core 2 Duo 1,66 GHz, RAM 1 GB, serta menggunakan *software* Matlab seri 7.8. Performa dari algoritma DE pada penelitian ini akan dibandingkan dengan algoritma CR(MC) yang dikembangkan oleh Rajendran (1995), algoritma HAMC (HAMC1, HAMC2, HAMC3) yang dikembangkan oleh Ravindran *et al* (2005) serta MOACSA (*Multi-Objective Ant Colony System Algorithm*) yang dikembangkan oleh Yagmahan dan Yenisey (2010).

4. HASIL PENELITIAN DAN ANALISA.

Parameter dari algoritma DE pada penelitian ini ditetapkan sebagai berikut : jumlah populasi = 20; $F_{min} = 0,5$; $Cr = 0,2-0,9$ dan jumlah iterasi sebanyak 200. Tiap pengujian algoritma akan direplikasi 10 kali dan hasil terbaik akan dipilih. Bobot dari

makespan dan *total flowtime* ditetapkan sebesar 0,5. Persentase relatif dari multi obyektif *makespan* dan *total flowtime* adalah sebagai berikut (Yagmahan dan Yenisey, 2010) :

$$RE = w_1 \cdot \left(\frac{C_{max} - \min C_{max}}{\min C_{max}} \right) + w_2 \cdot \left(\frac{F - \min F}{\min F} \right)$$

Persentase relatif dari obyektif *makespan*, *total flowtime* dan multi obyektif keduanya dapat dilihat pada tabel 1-3. Pada tabel tersebut terlihat bahwa algoritma DE yang dikembangkan pada penelitian ini memiliki performa yang lebih baik dibandingkan dengan algoritma CR(MC), HAMC maupun MOACSA.

Tabel 1. Performa dari algoritma untuk obyektif *makespan*

<i>n x m</i>	kasus	CR(MC)	HAMC1	HAMC2	HAMC3	MOACSA	DE
20 x 5	ta001	6,34	1,49	3,60	2,27	1,49	0,73
20 x 5	ta002	0,36	0,00	2,62	2,62	0,73	0,22
20 x 5	ta003	7,89	5,79	6,14	6,14	5,53	0,00
20 x 5	ta004	4,03	4,71	6,27	5,90	2,91	4,86
20 x 5	ta005	0,00	2,07	6,12	6,12	0,31	0,73
20 x 5	ta006	3,55	0,00	3,47	3,47	0,57	0,62
20 x 5	ta007	8,36	3,28	6,17	4,06	1,88	0,00
20 x 5	ta008	6,25	1,74	10,99	10,99	0,00	0,00
20 x 5	ta009	2,46	0,31	6,06	6,06	0,00	0,12
20 x 5	ta010	1,02	1,36	10,09	3,56	0,00	0,73
20 x 10	ta011	2,39	6,16	8,44	6,94	0,66	1,24
20 x 10	ta012	9,49	2,34	3,83	4,69	0,00	0,00
20 x 10	ta013	3,92	5,59	14,65	14,56	0,00	0,00
20 x 10	ta014	2,82	2,68	6,98	6,24	0,00	0,55
20 x 10	ta015	6,58	15,57	4,50	6,44	0,00	0,12
20 x 10	ta016	1,56	4,61	8,63	8,18	1,49	0,13
20 x 10	ta017	2,72	0,95	2,85	3,10	0,70	0,00
20 x 10	ta018	12,78	2,19	3,95	3,95	0,00	0,12
20 x 10	ta019	2,63	5,74	10,10	8,97	1,02	3,34
20 x 10	ta020	6,52	1,45	6,52	4,30	0,00	0,00
20 x 20	ta021	7,41	2,51	4,49	4,77	0,00	0,10
20 x 20	ta022	3,42	11,96	11,96	16,22	2,56	0,54
20 x 20	ta023	0,00	0,46	0,91	3,94	4,31	0,94
20 x 20	ta024	7,25	11,41	16,88	18,14	0,30	0,11
20 x 20	ta025	0,29	0,00	1,36	3,02	2,19	0,84
20 x 20	ta026	5,16	9,04	12,62	13,56	0,00	0,13
20 x 20	ta027	0,00	12,60	16,28	15,69	9,38	0,00
20 x 20	ta028	3,16	5,12	5,50	5,46	3,16	2,27
rata-rata		4,23	4,33	7,21	7,12	1,40	0,66

Tabel 2. Performa dari Algoritma untuk Obyektif *Total Flowtime*

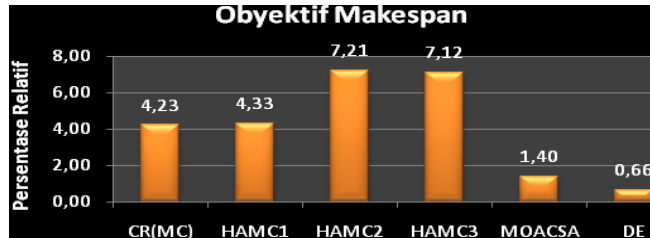
<i>n x m</i>	kasus	CR(MC)	HAMC1	HAMC2	HAMC3	MOACSA	DE
20 x 5	ta001	7,70	1,16	0,28	0,59	0,96	0,15
20 x 5	ta002	18,64	7,42	0,27	0,27	0,00	0,05
20 x 5	ta003	16,60	2,94	2,50	2,50	0,00	0,00
20 x 5	ta004	9,41	3,31	1,28	1,30	0,00	0,05
20 x 5	ta005	12,66	8,78	0,55	0,61	0,38	0,80
20 x 5	ta006	12,26	3,47	0,00	0,18	0,24	0,07

<i>n x m</i>	kasus	CR(MC)	HAMC1	HAMC2	HAMC3	MOACSA	DE
20 x 5	ta007	15,02	2,67	0,79	0,82	0,00	0,00
20 x 5	ta008	11,31	2,03	1,06	1,13	0,00	0,00
20 x 5	ta009	6,09	7,91	1,17	1,17	0,00	0,07
20 x 5	ta010	10,74	3,57	0,00	0,21	0,33	0,09
20 x 10	ta011	8,56	5,29	4,23	4,43	0,00	0,08
20 x 10	ta012	15,68	2,31	0,31	0,50	0,43	0,00
20 x 10	ta013	7,00	8,22	2,07	2,22	0,30	0,00
20 x 10	ta014	9,65	3,51	1,81	2,10	0,00	0,07
20 x 10	ta015	10,08	4,09	8,15	2,63	0,41	0,08
20 x 10	ta016	7,64	2,31	0,71	1,20	0,44	0,07
20 x 10	ta017	8,48	2,94	0,00	1,46	0,45	0,00
20 x 10	ta018	16,43	4,81	1,71	2,15	0,00	0,07
20 x 10	ta019	9,19	4,46	0,73	0,86	0,00	0,06
20 x 10	ta020	16,22	1,29	0,00	0,15	0,60	0,00
20 x 20	ta021	13,28	5,59	2,31	3,05	0,00	0,07
20 x 20	ta022	11,12	4,46	4,46	7,56	0,00	0,06
20 x 20	ta023	6,84	1,99	0,00	1,55	4,01	0,06
20 x 20	ta024	15,28	5,04	10,28	11,33	0,05	1,68
20 x 20	ta025	6,29	5,52	0,00	1,45	5,89	0,10
20 x 20	ta026	12,81	11,51	5,53	6,58	0,00	0,08
20 x 20	ta027	9,16	3,91	0,93	1,18	0,00	0,00
20 x 20	ta028	3,02	1,48	1,22	1,28	0,00	0,07
rata-rata		10,97	4,36	1,87	2,16	0,52	0,14

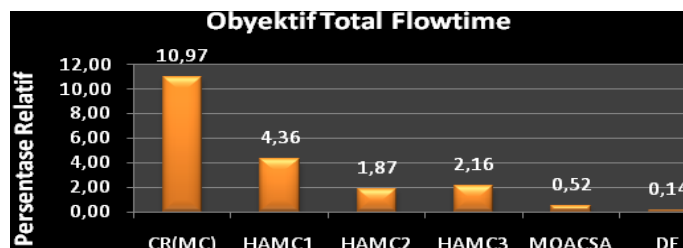
Tabel 3. Performa dari Algoritma untuk Multi Obyektif

<i>n x m</i>	kasus	CR(MC)	HAMC1	HAMC2	HAMC3	MOACSA	DE
20 x 5	ta001	7,58	1,19	0,56	0,73	1,53	0,40
20 x 5	ta002	17,04	6,72	0,38	0,38	0,00	0,36
20 x 5	ta003	14,56	1,95	1,57	1,57	0,05	0,00
20 x 5	ta004	8,65	3,10	1,36	1,35	0,00	0,08
20 x 5	ta005	11,35	7,99	0,85	0,90	0,67	0,10
20 x 5	ta006	11,19	2,87	0,00	0,16	0,12	0,10
20 x 5	ta007	14,14	2,43	0,96	0,81	0,00	0,00
20 x 5	ta008	10,64	1,76	1,64	1,70	0,00	0,00
20 x 5	ta009	5,30	6,79	1,10	1,10	0,00	0,10
20 x 5	ta010	9,57	3,04	0,49	0,15	0,00	0,10
20 x 10	ta011	7,94	5,18	4,36	4,44	0,00	0,10
20 x 10	ta012	14,59	1,74	0,00	0,23	0,19	0,00
20 x 10	ta013	6,20	7,45	2,41	2,55	0,00	0,00
20 x 10	ta014	8,75	3,07	1,80	2,02	0,00	0,11
20 x 10	ta015	9,54	4,66	7,61	2,65	0,33	0,12
20 x 10	ta016	7,06	2,35	1,17	1,58	0,40	0,10
20 x 10	ta017	7,80	2,56	0,00	1,37	0,38	0,00
20 x 10	ta018	15,64	4,15	1,41	1,82	0,00	0,10
20 x 10	ta019	8,23	4,11	1,00	1,04	0,00	0,09
20 x 10	ta020	14,97	0,84	0,03	0,00	0,37	0,00
20 x 20	ta021	12,80	5,31	2,38	3,08	0,00	0,09
20 x 20	ta022	10,28	4,64	4,64	7,88	0,00	0,12
20 x 20	ta023	6,31	1,83	0,00	1,65	4,27	1,20
20 x 20	ta024	14,53	5,27	10,52	11,58	0,35	0,10
20 x 20	ta025	5,78	5,04	0,00	1,47	5,69	2,35
20 x 20	ta026	12,26	11,30	5,96	7,00	0,00	0,11
20 x 20	ta027	7,85	3,73	1,18	1,37	0,00	0,00
20 x 20	ta028	2,81	1,50	1,29	1,34	0,00	0,09
rata-rata		10,12	4,02	1,95	2,21	0,51	0,21

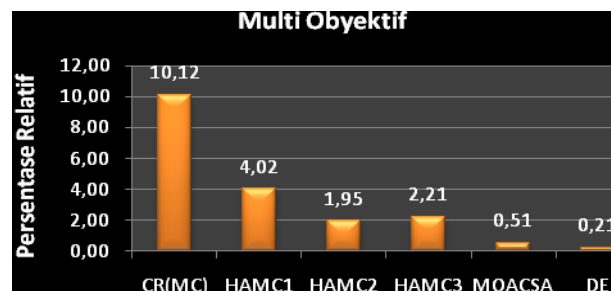
Gambar 1-3 menunjukkan rata-rata performa dari masing-masing algoritma untuk obyektif *makespan*, *total flowtime* maupun multi obyektif. Pada gambar tersebut terlihat bahwa algoritma DE pada penelitian ini memiliki performa yang lebih baik.



Gambar 1. Rata-rata persentase relatif obyektif *makespan*



Gambar 2. Rata-rata persentase relatif obyektif *total flowtime*



Gambar 3. Rata-rata persentase relatif multi obyektif

5. KESIMPULAN

Hasil dari penelitian ini menunjukkan bahwa algoritma DE yang dikembangkan pada penelitian ini mempunyai performa yang lebih baik dibandingkan dengan algoritma CR(MC), HAMC1, HAMC 2, HAMC3 maupun MOACSA untuk obyektif *makespan*, *total flowtime* maupun multi obyektif keduanya. Dari gambar 1-3 dapat dilihat bahwa algoritma DE yang dikembangkan pada penelitian ini sangat dominan lebih baik jika dibandingkan dengan algoritma CR(MC), HAMC1, HAMC2 dan HAMC3 serta MOACSA untuk obyektif *makespan*, *total flowtime* dan multi obyektif. Nilai rata-rata persentase relatif dari algoritma DE selalu dibawah 1%.

Algoritma DE pada riset ini juga dapat diuji pada obyektif tunggal maupun multi obyektif dengan mempertimbangkan obyektif-obyektif yang lain seperti *mean flowtime*, *total tardiness* maupun *maximum tardiness*. Kedepannya, DE dengan *adaptive parameters* dan *local search* pada riset ini dapat juga diaplikasikan pada permasalahan

penjadwalan pada beberapa sistem manufaktur yang lain seperti *job shop*, *cellular manufacturing* dan *flexible manufacturing*.

6. DAFTAR RUJUKAN

- Baker, Kenneth R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons, Inc.
- Mingyong, L. dan Erbao, C. (2010), "An Improved Differential Evolution Algorithm for Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows", *Engineering Applications of Artificial Intelligence*, Vol. 23, hal. 188–195.
- Pan, Q.K., Tasgetiren, M.F. dan Liang, Y.C. (2008), "A Discrete Differential Evolution Algorithm for The Permutation Flowshop Scheduling Problem", *Computers & Industrial Engineering*, Vol. 55, hal. 795–816.
- Qian, B., Wang, L., Huang D., Wang, W.L. dan Wang, X. (2008), "A Hybrid Differential Evolution Method for Permutation Flow-Shop Scheduling", *The International Journal of Advanced Manufacturing Technology*, Vol. 38, hal. 757–777.
- Rajendran, C. (1995). "Heuristics for scheduling in flowshop with multiple objectives", *European Journal of Operational Research*, Vol. 82, hal. 540–555.
- Ravindran, D., Noorul Haq, A., Selvakumar, S. J. dan Sivaraman, R. (2005), "Flow Shop Scheduling With Multiple Objective of Minimizing Makespan and Total Flow Time", *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 9-10, hal. 1007–1012.
- Seda, M. (2007), "Mathematical Models of Flow Shop and Job Shop Scheduling Problems", *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 31, hal. 122-127.
- Storn, R. dan Price, K. (1997), "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Space", *Journal of Global Optimization*, Vol. 11, hal. 341-359.
- Taillard, E. (1993), "Benchmarks for Basic Scheduling Problems", *European Journal of Operational Research*, Vol. 64, No. 2, hal. 278-285.
- Tvrđik, J. (2006), "Differential Evolution: Competitive Setting of Control Parameters", *Proceedings of the International Multiconference on Computer Science and Information Technology*, University of Ostrava, Ostrava, hal. 207–213.
- Yagmahan, B. dan Yenisey, M. M. (2010), "A Multi-Objective Ant Colony System Algorithm for Flow Shop Scheduling Problem", *Expert Systems with Applications*, Vol. 37, No. 2, hal. 1361–1368.