

PENYELESAIAN MASALAH PENJADWALAN FLOWSHOP DENGAN *MODIFIED GENETIC ALGORITHM*

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹Program Studi D3 Sistem Informasi Fakultas Vokasi

²Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi
Universitas Airlangga

E-mail: dyahherawatie@gmail.com, etowuryanto@gmail.com dan nasazata@gmail.com

Abstrak

Flowshop Scheduling Problem (FSP) merupakan salah satu *timetabling/scheduling problem*. Masalah ini merupakan masalah perencanaan berbasis waktu dan optimasi kombinatorial. Penjadwalan *flowshop* merupakan sebuah permasalahan perencanaan produksi dengan *n-job* (item, tugas, dan lain-lain) yang harus diproses dalam urutan yang sama pada setiap *m*-mesin. Masing-masing *job* mempunyai *processing time* yang berbeda untuk mesin yang berbeda. Tujuan dari penjadwalan ini adalah untuk mendapatkan urutan dari *job-job* untuk meminimalkan *makespan*. Salah satu metode heuristik yang sering digunakan adalah Algoritma genetika (Genetic Algorithm atau GA). Algoritma ini merupakan yang cenderung diselesaikan dengan kerjasama dari pencarian dan heuristik, yang mengarah pada solusi yang memuaskan tetapi sub – optimal. *Modified Genetic Algorithm* (MGA)) merupakan algoritma genetika yang skema seleksi, crossover, dan strategi mutasinya telah dimodifikasi. MGA ini bertujuan untuk menghindari optimal lokal, dan menemukan solusi optimal secara efisien. Penelitian ini bertujuan untuk membandingkan keefektifan metode MGA dalam menyelesaikan masalah *flowshop scheduling*. Untuk penelitian ini digunakan seleksi elitism, *partial schedule exchange crossover*, dan *simple inversion mutation* (SIM). Dengan menggunakan tiga set data, ditunjukkan bahwa MGA lebih efisien dalam menentukan himpunan solusi layak daripada GA.

Kata kunci: *Timetabling Problem, Flowshop Scheduling Problem, Modified Genetic Algorithm*

1. PENDAHULUAN

Penjadwalan merupakan salah satu bagian penting dalam bidang industri, terutama di bagian manufaktur dan produksi. Permasalahan penjadwalan di bidang produksi meliputi pengaturan *job-job* yang akan diproses pada serangkaian mesin dengan urutan *job* yang sama berlaku untuk setiap mesin dan setiap mesin hanya memproses sebuah *job* pada saat yang sama. Masalah ini dikenal dengan istilah *flowshop*. Permasalahan utama pada *flowshop* adalah menentukan urutan *job* yang akan dipertahankan di sepanjang sistem yang memenuhi fungsi tujuan.

Permasalahan penjadwalan *flowshop* diperkenalkan oleh S.M Johnson pada tahun 1954 dengan permasalahan yang dikemukakan berupa permasalahan penjadwalan *flowshop* 2-mesin dengan fungsi tujuan meminimumkan *makespan*.

Beberapa algoritma yang bisa digunakan untuk menyelesaikan permasalahan penjadwalan *flow shop m*-mesin, antara lain algoritma Palmer, Gupta, CDS (Cambell, Dudek, Smith), dan algoritma heuristik RA (*Rapid Access*). Algoritma ini mempunyai fungsi tujuan meminimumkan *makespan*.

Metode lain yang dapat digunakan untuk memperoleh solusi dalam persoalan *Flowshop Scheduling* adalah algoritma genetik. Algoritma genetika (Genetic Algorithm atau GA) adalah salah satu algoritma heuristik untuk menyelesaikan masalah kombinatorik yang terinspirasi teori evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu-individu yang kuat yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹Program Studi D3 Sistem Informasi Fakultas Vokasi

²Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

Universitas Airlangga

pada individu melalui proses perkembang-biakan. Dalam algoritma genetika ini, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: "Bagaimana mendapatkan keturunan yang lebih baik". Operator-operator yang digunakan dalam GA antara lain, seleksi, mutasi dan perkawinan silang (crossover). Sedangkan algoritma genetika yang dimodifikasi (*Modified Genetic Algorithm (MGA)*) merupakan algoritma genetika yang skema seleksi, crossover, dan strategi mutasinya telah dimodifikasi. *MGA* ini bertujuan untuk menghindari optimum lokal, dan menemukan solusi optimal secara efisien (Modupe, dkk, 2014).

Penelitian ini bertujuan untuk membandingkan keefektifan metode *MGA* dalam menyelesaikan masalah *flowshop scheduling*. Untuk penelitian ini digunakan seleksi elitism, *partial schedule exchange crossover*, dan *simple inversion mutation (SIM)*. Pengkodean permutasinya menggunakan representasi berdasarkan operasi menurut Gen, Tsujimura, dan Kubota (Gen dan Cheng, 1997). Dengan jenis pengkodean permutasi menggunakan representasi tersebut akan langsung didapat individu yang fisibel karena gen-gen yang didapat secara random diurutkan berdasarkan urutan operasi dan mesin sesuai kasus.

1. KAJIAN LITERATUR

Penjadwalan *Flowshop*

Seperti dijelaskan oleh Sahu (2009), penjadwalan *flowshop* sejenis dengan masalah kombinatorial. Penjadwalan *flowshop* merupakan sebuah permasalahan perencanaan produksi dengan *n-job* (item, tugas, dan lain-lain) yang harus diproses dalam urutan yang sama pada setiap *m*-mesin.. Masing-masing *job* mempunyai *processing time* yang berbeda untuk mesin yang berbeda. Beberapa karakteristik dari penjadwalan *flowshop*, antara lain :

- a. Terdapat *m* mesin dan *n* job

- b. Masing-masing *job* terdiri dari *m* operasi dan masing-masing operasi membutuhkan mesin yang berbeda
- c. Ke-*n* *job* diproses dalam urutan yang sama pada *m* mesin
- d. Waktu pemrosesan dari *job* ke-*i* pada mesin ke-*j* dinotasikan dengan t_{ij} ($i = 1, 2, \dots, n$, dan $j = 1, 2, \dots, m$)
- e. Disusun sebuah jadwal berupa urutan *job*, yang akan memenuhi tujuan tertentu. Tujuan yang sering digunakan adalah meminimumkan *makespan*.

Dalam penjadwalan *flowshop* digunakan asumsi antara lain :

- a. Setiap *job* diproses pada semua mesin berdasarkan urutan tertentu
- b. Setiap mesin hanya memproses satu *job* pada suatu waktu
- c. Setiap *job* diproses pada satu mesin pada suatu waktu
- d. Operasi tidak pre-emptif
- e. Waktu set-up untuk sebuah operasi adalah *sequence-independent* dan tidak termasuk dalam waktu pemrosesan.

Algoritma Genetik (GA)

Algoritma Genetik (*Genetic Algorithm (GA)*) merupakan merupakan sebuah kelompok dari metode-metode untuk menyelesaikan permasalahan-permasalahan dengan menggunakan algoritma yang diinspirasi oleh proses-proses teori evolusi baru (Fang, 1994). Langkah pertama pada algoritma genetik adalah menerjemahkan / merepresentasikan masalah riil menjadi terminologi biologi. Cara untuk merepresentasikan masalah ke dalam bentuk kromosom disebut pengkodean. Terdapat beberapa cara pengkodean seperti pengkodean biner, permutasi, nilai, dan pohon. Pemilihannya berdasarkan masalah yang dihadapi.

Operator-operator yang digunakan dalam GA adalah sebagai berikut: (Mawaddah dan Mahmudy, 2006)

- a. Seleksi

Proses ini bertanggung jawab untuk melakukan pemilihan terhadap individu yang hendak diikuti dalam proses reproduksi. Langkah pertama yang dilakukan dalam seleksi ini

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹⁾Program Studi D3 Sistem Informasi Fakultas Vokasi

²⁾Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

Universitas Airlangga

- adalah pencarian nilai fitness. Seleksi mempunyai tujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang mempunyai nilai *fitness* terbaik. Beberapa metode seleksi antara lain; roda *roulette*, rangking, dan turnamen.
- b. Perkawinan silang (crossover)
Proses ini berfungsi untuk menghasilkan keturunan dari dua buah kromosom induk yang terpilih. Kromosom anak yang dihasilkan merupakan kombinasi gen-gen yang dimiliki oleh kromosom induk.
- c. Mutasi
Proses ini dilakukan dengan cara melakukan perubahan pada sebuah gen atau lebih dari sebuah individu. Tujuan dari mutasi adalah agar individu-individu yang ada dalam populasi semakin bervariasi. Mutasi akan sangat berperan jika pada populasi awal hanya ada sedikit solusi yang mungkin terpilih. Sehingga, operasi itu sangat berguna dalam mempertahankan keanekaragaman individu dalam populasi meskipun dengan mutasi tidak dapat diketahui apa yang terjadi pada individu baru.

Algoritma genetika secara garis besar data dijabarkan sebagai berikut (Wan dan Birch, 2011) :

- 1) Satu set n pengaturan dari regressors dengan menggunakan operasi seleksi, crossover, dan mutasi dari x_1, \dots, x_n untuk mendapatkan $x_1^*, x_2^*, \dots, x_n^*$
- 2) Mengevaluasi Menentukan fungsi tujuan/fitness, dan variabel-variabelnya. Tentukan operasi GA (seperti ukuran populasi, rasio *parent/offspring*, metode seleksi, banyaknya *crossover* dan tingkat mutasi) .
- 3) Membangkitkan secara acak populasi awal sebanyak n individu, yaitu x_1, x_2, \dots, x_n .
- 4) Mengevaluasi setiap kromosom dalam populasi awal dengan fungsi tujuan, dengan menghitung $D(x_1), D(x_2), \dots, D(x_n)$. di mana $D(x_i)$ adalah fungsi

keinginan yang dievaluasi pada kromosom x_i .

- 5) Menghasilkan populasi *offspring* dengan operasi GA (seperti seleksi, crossover, dan mutasi), yaitu menghasilkan setiap individu dalam populasi *offspring* dengan menggunakan fungsi tujuan.
- 6) Tentukan individu-individu yang akan dimasukkan dalam populasi berikutnya. Langkah ini disebut dengan "*replacement*". Individu dari populasi *parent* saat ini diganti oleh individu yang berasal dari populasi *offspring* dan/ atau *parent*.
- 7) Jika kriteria pengentian dipenuhi, maka prosedur dihentikan. Jika tidak, kembali ke langkah 4.

Modified Genetic Algorithm (MGA)

Prosedur MGA sama dengan GA, kecuali bahwa pada generasi ke- i antara langkah 5 dan 6 ditambahkan langkah D sebagai berikut (Wan dan Birch, 2011):

- d. Apakah kromosom terbaik pada populasi *offspring* juga terbaik untuk populasi *parent* saat ini dan apakah *offspring* terbaik memiliki nilai $D > D_{\text{cutoff}}$? Apakah $\max(D(x_i^*)) > D_{\text{cutoff}}$?
- d-1 . Jika tidak, lanjutkan ke Langkah 6 .
- d-2. Jika ya, maka tentukan dan lakukan pencarian kromosom lokal sebagai titik awal. Pencarian lokal berakhir ketika fungsi obyektif tidak meningkat. Cari kromosom dengan nilai keinginan terbesar dan mengganti kromosom terbaik dengan nilai keinginan terbesar dalam populasi *offspring*. Lalu lanjutkan ke Langkah 6 .

2. METODE PENELITIAN

Langkah-langkah yang dilakukan untuk mendapatkan penyelesaian dari masalah penjadwalan flowshop dengan *Modified Genetic Algorithm* adalah sebagai berikut :

- a. Melakukan penelusuran, penelaahan literatur, serta diskusi yang intensif, yang membahas tentang penjadwalan flowshop yang berhubungan dengan *Genetic Algorithm* dan *Modified Genetic Algorithm*.

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹⁾Program Studi D3 Sistem Informasi Fakultas Vokasi

²⁾Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

Universitas Airlangga

- b. Menyusun algoritma penjadwalan flowshop dengan *Genetic Algorithm* (GA).
- c. Menyusun algoritma penjadwalan flowshop dengan *Modified Genetic Algorithm* (MGA).
- d. Mengimplementasikan kedua algoritma tersebut ke dalam program bahasa PHP.
- e. Membandingkan keefektifan algoritma dengan menggunakan data sekunder. Untuk membandingkan keefektifan kedua algoritma digunakan nilai minimum, maximum, mean and varians dari makespan.

Operator-operator yang digunakan dalam GA maupun MGA pada penelitian ini adalah *partial schedule of exchange crossover, simple inverse mutation* dan seleksi rod roulette. Pengkodean digunakan untuk merepresentasikan kromosomnya adalah pengkodean permutasi, yang berisi urutan bilangan integer yang menunjukkan urutan penyelesaian job. Untuk mengevaluasi kinerja GA dan MGA digunakan tiga set data. Data pertama adalah data 5 job 3 mesin dari Rajendran (1995), data kedua data 10 job 5 mesin dari Kattan (2003), dan data ketiga data 10 job 10 mesin dari Sahu (2009). Untuk penghentian iterasi, digunakan kriteria maksimum generasi (max.gen).

Tabel 1. Data (5×3) dari Rajendran (1995)

	M0	M1	M2
J0	5	10	9
J1	2	3	7
J2	7	9	3
J3	3	2	18
J4	4	3	9

Tabel 2. Data (10×5) mesin dari Kattan (2003)

	M0	M1	M2	M3	M4
J0	5	12	15	4	13

J1	10	8	6	9	4
J2	13	17	5	8	10
J3	21	13	18	14	17
J4	15	6	9	4	13
J5	14	10	3	7	22
J6	12	5	6	9	14
J7	13	7	4	8	3
J8	10	14	13	20	5
J9	6	19	14	13	18

Tabel 3. Data (10×10) dari Sahu (2009)

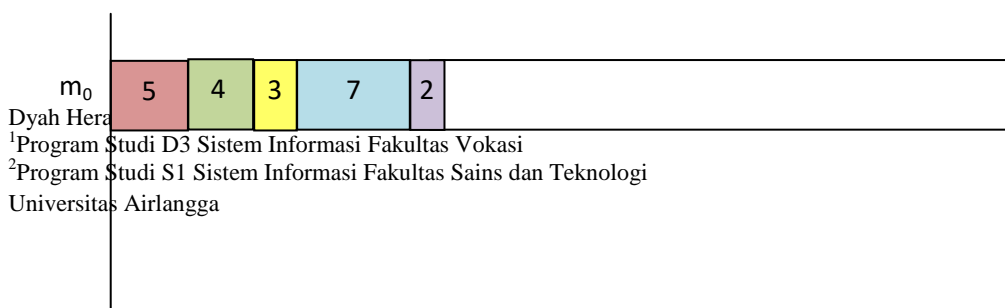
	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
J0	5	2	3	5	7	9	7	8	2	7
J1	2	6	4	2	6	2	5	2	6	1
J2	1	2	2	1	3	7	2	5	4	4
J3	7	5	6	3	2	3	2	4	2	2
J4	6	6	1	8	6	4	3	9	6	4
J5	3	7	5	2	2	1	5	3	2	6
J6	7	2	4	6	5	5	1	2	5	2
J7	5	1	7	1	7	3	6	6	2	2
J8	7	8	6	9	1	8	2	1	6	6
J9	4	3	5	8	3	1	3	8	3	7

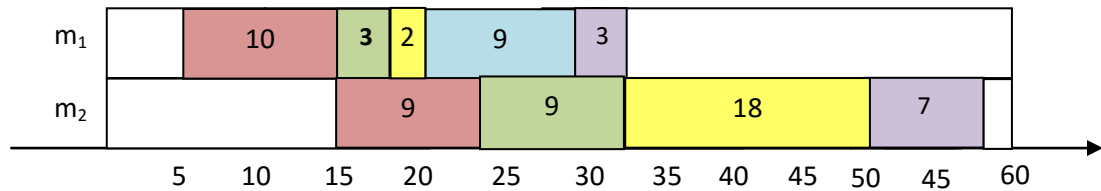
2.1. Representasi Kromosom

Dalam GA maupun MGA, proses *generate* populasi awal dimulai dengan melakukan *generate* kromosom dengan *generate* gen-gen secara acak. Pada setiap kromosom sejumlah *job* diacak sebanyak mesin, kemudian menjadi anggota dalam populasi awal. Proses ini dilakukan sebanyak *pop_size*. Dari kromosom yang terbentuk akan ditentukan operasinya dengan mengurutkan operasi-operasinya berdasarkan urutan *job*nya. Dengan menggunakan data set yang pertama, misalnya digenerate kromosom sebagai berikut :

<i>Job</i>	0	4	3	2	1
------------	---	---	---	---	---

Mengevaluasi masing-masing kromosom dengan menghitung bobot masing-masing kromosom. Adapun evaluasinya dengan membentuk *machine gantt chart*, contoh pada kromosom seperti di atas dapat dilihat pada gambar 1.





Gambar 1. Machine gantt chart

Keterangan :

- t_{i1} Waktu proses untuk job-0
- t_{i2} Waktu proses untuk job-1
- t_{i3} Waktu proses untuk job-2
- t_{i4} Waktu proses untuk job-3
- t_{i5} Waktu proses untuk job-4

digunakan, masing-masing data 5 job 3 mesin, 10 job 5 mesin, dan 10 job 10 mesin. Ketiganya mewakili data berukuran kecil, sedang, dan besar seperti ditunjukkan pada tabel 1, 2, dan 3. Untuk setiap set data, masing-masing akan digunakan ukuran populasi (pop.size) sebesar 15, dan 20, maksimum generasi (max gen) sebesar masing-masing 10, 30, dan 60, *crossover rate* (p_c) = 0,6, *mutation rate* (p_m) = 0,01. Untuk setiap set data, dengan pop.size dan maxgen tertentu, akan dilakukan percobaan dengan diulang sebanyak 10 kali. Dari masing-masing percobaan, akan dilihat nilai makespan. Masing-masing nilai minimum, maksimum, means, standard deviasi. Hasilnya bisa dilihat pada tabel 4.

3. HASIL DAN PEMBAHASAN

Untuk mengevaluasi kinerja GA dan MGA, tiga kelompok set data

Tabel 4. Perbandingan nilai makespan hasil dari GA dan MGA

	pop.size	nilai	GA			MGA		
			Max Gen			Max Gen		
			10	30	60	10	30	60
Data 1	15	min	51	51	51	51	51	51
		max	51	51	51	51	51	51
		mean	51	51	51	51	51	51
		stdev	0	0	0	0	0	0
	20	min	51	51	51	51	51	51
		max	51	51	51	51	51	51
		mean	51	51	51	51	51	51
		stdev	0	0	0	0	0	0
Data 2	15	min	171	162	160	158	161	158
		max	173	177	171	165	167	163
		mean	172,6	171,1	168,2	163,1	163,9	161,6
		stdev	0,84	5,78	3,39	2,13	1,6	1,51
	20	min	167	161	158	161	161	158
		max	169	171	171	166	164	162
		mean	168,1	168,4	165,6	163,5	162,2	160,8
		stdev	0,57	3,17	5,42	1,51	1,03	1,23

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹⁾Program Studi D3 Sistem Informasi Fakultas Vokasi

²⁾Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

Universitas Airlangga

Data 3	15	min	95	95	92	91	91	91
		max	101	99	100	95	93	93
		mean	97,5	97,3	96,7	92,6	92,1	92
		stdev	3,03	1,49	2,54	1,07	0,57	0,67
	20	min	95	92	91	91	92	91
		max	99	98	94	94	93	93
		mean	96,9	94,3	92,5	92,8	92,2	92
		stdev	2,02	1,89	0,97	1,03	0,42	0,47

Dari hasil percobaan, untuk data kecil (5x3), metode MGA memberikan hasil yang seragam jika dibandingkan dengan GA untuk semua ukuran performans. Urutan ob dengan minimum makespan sebesar 51 antara lain sebagai berikut: 3, 1, 2, 0, 4, atau 3, 4, 2, 0, 1, atau 3, 1, 4, 0, 2, dan 1, 3, 2, 4, 0.

Jika dilihat dari mean makespan untuk data kedua dan ketiga menunjukkan pola yang hampir sama. Hal ini bisa dilihat pada tabel 4, gambar 1 dan gambar 3. Jika dibandingkan nilai mean makespannya, Baik MGA pop.size 15 maupun 20 cenderung memberikan nilai makespan yang lebih kecil. Pada metode GA, dengan menambah maxgen, cenderung diperoleh nilai makespan yang lebih kecil.

Pada data kedua, diperoleh nilai minimum makespan sebesar 158. Jika dibandingkan dengan penelitian sebelumnya (Herawatie, dan Wuryanto, 2012), diperoleh minimum makespan sebesar 161. Sedangkan pada data ketiga, dengan menggunakan MGA diperoleh nilai minimum makespan sebesar 91, Sedangkan penelitian terdahulu menghasilkan minimum makespan sebesar 93. Dengan menggunakan kedua

data, rata-rata MGA memberikan hasil yang lebih baik. Untuk data kedua urutan job terbaik (dengan maksepan 158) adalah 0, 5, 6, 9, 3, 8, 4, 2, 1, 7 atau 0, 5, 6, 9, 3, 8, 4, 2, 7, 1. Sedangkan Data ketiga urutan job terbaik (dengan makespan 91) adalah 2, 0, 5, 1, 4, 9, 8, 7, 3, 6 atau 2, 1, 4, 9, 5, 0, 8, 7, 6, 3.

Jika dilihat dari nilai standart deviasinya, data kedua dan ketiga juga menunjukkan pola yang hampir sama (gambar 2 dan 4). Metode MGA cenderung memberikan nilai standart deviasi yang lebih kecil dibandingkan GA. Hal ini menunjukkan nilai makespan yang dihasilkan lebih seragam.

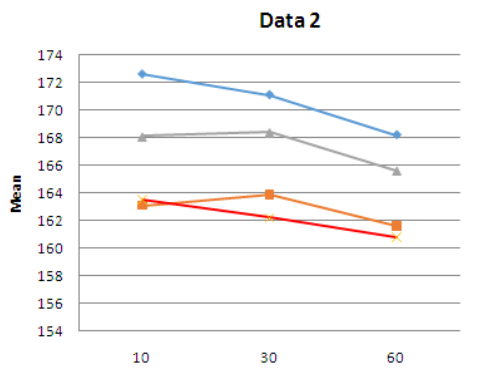
Percobaan yang dilakukan pada penelitian ini, mendukung mendukung penjelasan dari Modupe, dkk (2014) bahwa salah satu kelemahan dari GA adalah ditemuinya optimum lokal. Optimum lokal diperoleh karena adanya konvergensi dini yang disebabkan karena kurangnya diversitas populasi setelah melewati sekian generasi. Meskipun MGA memberikan hasil yang lebih memuaskan, tetapi dalam prakteknya komputasi MGA membutuhkan waktu yang lebih lama.

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

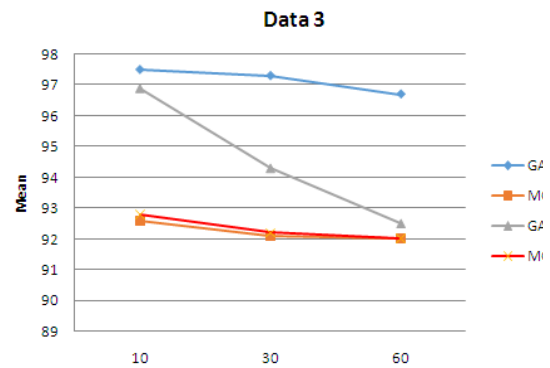
¹⁾Program Studi D3 Sistem Informasi Fakultas Vokasi

²⁾Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

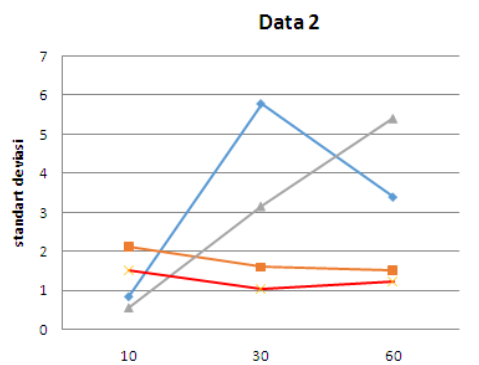
Universitas Airlangga



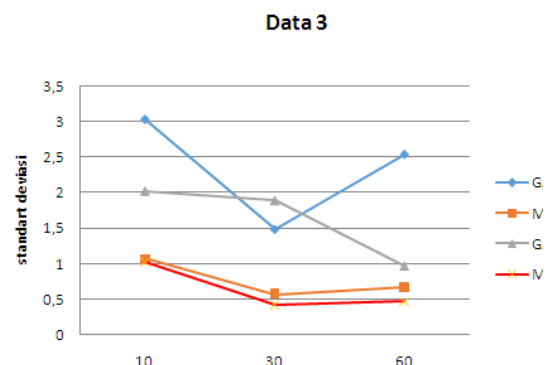
Gambar 1. Mean makespan untuk data kedua



Gambar 3. Mean makespan untuk data ketiga



Gambar 2. Standart deviasi makespan untuk data kedua



Gambar 4. Standart deviasi makespan untuk data ketiga

4. KESIMPULAN

Berdasarkan perbandingan nilai minimum, maksimum, mean, dan standart deviasi dari nilai makespan dengan menggunakan ketiga set data dapat disimpulkan antara lain:

- Untuk data berukuran kecil, dengan menggunakan MGA maupun GA memberikan hasil yang sama.
- Metode MGA cenderung menghasilkan solusi yang lebih baik pada data berukuran sedang maupun besar, dibandingkan dengan metode GA untuk semua indikator.
- Metode MGA bisa mencegah diperolehnya solusi optimum lokal. Optimum lokal diperoleh karena terjadinya konvergensi dini.

5. REFERENSI

- [1]. Champbell, H.G., Dudek, R.A, and Smith, M.L, 1970. *A Heuristic Algorithm for the n-job, m-machine sequencing problem*, Management Science 16, p. B630-B637.
- [2]. Fang H-L., 1994, *Genetic Algorithms in Timetabling and Scheduling*, Department of Artificial Intelligence University of Edinburgh.
- [3]. Gen, M. dan Cheng, R., 1997, *Genetic Algorithms and Engineering Design*, John Wiley and Sons, New York.
- [4]. Herawatie, Dyah dan Eto Wuryanto, 2012. Penjadwalan Flowshop dengan Metode Heuristik Multiple Objective Terboboti, *Seminar Nasional Teknologi Informasi & Komputasi (SENASTIK)*, Universitas Trunojoyo Madura.
- [5]. Kattan, Ibrahim, Mikolajczak, Boleslaw, Kattan, Khalid,

Dyah Herawatie^{1,2)}, Eto Wuryanto²⁾, Nasa Zata Dina¹⁾

¹⁾Program Studi D3 Sistem Informasi Fakultas Vokasi

²⁾Program Studi S1 Sistem Informasi Fakultas Sains dan Teknologi

Universitas Airlangga

- Alqassar, Bassam, 2003. Minimizing Cycle Time and Group Scheduling, Using Petri Nets A Study of Heuristic Methods, *Journal of Intelligent Manufacturing*, Vol. 14, p 107-121.
- [6]. Mawaddah NK and Wayan Firdau Mahmudy, 2006. *Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika*, *Kursor*, vol 2, no.2, pp.1-8.
- [7]. Modupe, A. O., Omidiora E. Olusayo, dan Olabiyisi S. Olatunde, 2014. *Development of a University Lecture Timetable using Modified Genetic Algorithms Approach*, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 4, Issue 9, September 2014.
- [8]. Rajendran, Chandrasekharan 1995. Heuristic for Scheduling in Flowshop with Multiple Objectives, *European Journal of Operational Research*, Vol.82, p 540-555,
- [9]. Ravindran, D., Haq, A. Noorul, Selvakuar, S.J., Sivaraman, R. 2005. Flow Shop Scheduling With Multiple Objective of Minimizing Makespan and Total Flow Time, *Int J Adv Manuf Technol*, Vol. 25, p 1007-1012.
- [10]. Sahu, Atul Kumar, 2009. *Efficient Heuristics for Scheduling Tasks on A Flow Shop Environment to Optimize Makespan*, Thesis, Departemen of Mechanical Engineering, National Institute of Technology, Rourkela.
- [11]. Wan, Wen, dan Jeffrey B. Birch, 2011. *Using a modified genetic algorithm to find feasible regions of a desirability function*. *Quality and Reliability Eng. Int.* 27(8): 1173-1182.