

PENJADWALAN *FLOW SHOP* DENGAN PENDEKATAN *CROSS ENTROPY-GENETIC ALGORITHM* UNTUK MENURUNKAN *MAKESPAN* PADA PEMBUATAN RODA GIGI

Dian Setiya Widodo, Mario Sarisky Dwi Ellianto

Politeknik 17 Agustus 1945 Surabaya
diansetiawidodo@yahoo.com

Abstrak

Flow shop adalah proses penentuan urutan pekerjaan yang memiliki jalan produk yang sama. Sementara penjadwalan *flow-shop* mengambil asumsi bahwa sejumlah pekerjaan yang masing-masing memiliki urutan kerja mesin yang sama. Masalah-masalah yang dihadapi oleh perusahaan, yaitu tingginya permintaan padahal belum ada perencanaan penjadwalan yang baik akibatnya perusahaan harus mampu mengoptimalkan penjadwalan pekerjaan. Salah satu cara untuk memecahkan masalah ini adalah dengan meminimalkan *Makespan*. Penelitian ini akan memecahkan masalah penjadwalan dengan menggunakan *cross entropi-genetik algorithm* (CEGA) sebuah metode yang digunakan untuk meminimalkan *Makespan*. Hasil yang diberikan oleh CEGA selanjutnya dibandingkan dengan hasil dari metode yang selama ini diterapkan di perusahaan. Untuk mendukung aplikasi CEGA digunakan perangkat lunak MATLAB untuk. Ternyata hasil penelitian bisa memberikan solusi optimal, nilai-nilai *Makespan* yang diperoleh untuk 10829 detik dan lebih efektif bila dibanding dengan metoda yang dipakai perusahaan dengan efisiensi *Makespan* 10.06%.

Kata kunci: *flow shop*, penjadwalan, *cross entropi-genetic algorithm*, *Makespan*.

Abstract

Flow shop is the process of determining the sequence of jobs that have the same product path. While the *flow-shop* scheduling takes assumption that a number of jobs that each has the same machine work sequence. The problems faced by companies, namely the high demand and yet the existence of a good scheduling planning resulted in the company should be able to optimize scheduling job. One of the ways to solve these problems is by minimizing the *Makespan*. This research will solve the problem regarding to the *flow shop* scheduling by using *cross entropy-genetic algorithm* (CEGA) method to minimize *Makespan*. The technique that is used to solve the problem is by comparing the result of the application of existing methods in the company with the proposed method (CEGA). To support the application of CEGA used MATLAB software. Finally known that the results CEGA can give optimal solution, *Makespan* values obtained for 10829 seconds. So far, it was more effective than the method that used at the firm with *Makespan* efficiency by 10.06%.

Keywords: *Flow Shop*, Scheduling, *Cross Entropy-Genetic Algorithm*, *Makespan*.

PENDAHULUAN

Salah satu usaha yang dilakukan untuk tercapainya penjadwalan yang optimal adalah dengan meminimalkan *makespan*. *Makespan* merupakan aktivitas proses dengan meminimalkan total waktu penyelesaian serangkaian pekerjaan (*job*). Munculnya permasalahan penjadwalan terjadi apabila pada saat yang sama terdapat sekumpulan pekerjaan yang harus dikerjakan dengan terbatasnya mesin atau fasilitas produksi yang tersedia (Baker, 1974). Penjadwalan *flow shop* (*flow shop scheduling*) merupakan penjadwalan produksi yang setiap pekerjaan akan melalui setiap mesin dengan urutan yang sama, dengan salah satu tujuannya yaitu untuk menyelesaikan serangkaian pekerjaan berdasarkan urutan proses dengan tujuan untuk mendapatkan nilai *makespan* yang optimal.

PT.X merupakan perusahaan yang bergerak pada produksi roda gigi. Dalam proses pembuatan roda gigi ini aliran proses produksi yang dipakai yaitu model *flow shop*. Permasalahan yang dihadapi perusahaan yaitu tingginya permintaan dan belum adanya perencanaan penjadwalan roda gigi yang mengakibatkan perusahaan harus dapat mengoptimalkan penjadwalan pekerjaan, untuk dapat meminimasi *makespan* yang nantinya diharapkan mampu meningkatkan jumlah produksi dan kebutuhan komponen tersebut dapat terpenuhi.

Metaheuristik merupakan metode pendekatan yang didasarkan pada metode *heuristic* yaitu suatu metode untuk mencari solusi dengan pendekatan komputasi yang memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik lokal optimum dan melakukan pencarian diruang solusi untuk menemukan solusi global dengan cara mencoba-coba secara iteratif untuk memperbaiki kandidat solusi yang diinginkan (Santosa dan Willy, 2011). Metode pendekatan *metaheuristik* ini dapat dipakai untuk mengatasi permasalahan penjadwalan tersebut.

Metode *metaheuristik* salah satu diantaranya yaitu algoritma *cross entropy* dan *genetic algorithm*. Metode algoritma *Cross Entropy* merupakan teknik yang cukup baru tujuannya adalah untuk menghasilkan urutan solusi yang memusat dengan cepat kearah solusi yang optimal dari hasil iterasi. Metode *Cross Entropy* Awalnya diterapkan untuk simulasi kejadian langka (*rare-event*), lalu dikembangkan untuk beberapa kasus seperti optimasi kombinatorial, optimasi kontinyu, *machine learning*, dan beberapa kasus lain (Santosa, dan Willy, 2011). Algoritma *Cross Entropy* ini memiliki cara kerja sebagai berikut: strategi perolehan informasi oleh algoritma ini adalah bagaimana mengambil sampel acak yang tepat dari ruang lingkup permasalahan dengan mendapatkan gambaran distribusi dari solusi yang bagus. Distribusi ini akan terus *di-update* berdasarkan kandidat solusi yang lebih baik. Selanjutnya sampel akan dibangun berdasarkan rata-rata distribusi yang muncul dari solusi yang bagus. Algoritma akan terus mengulang skenario yang sama hingga distribusi sampel mengarah pada area solusi optimal (Nurkhalida, dan Santosa, 2012)

Sedangkan Algoritma Genetika merupakan teknik pencarian yang didasarkan pada mekanisme seleksi alam dan genetika alami. Genetik Algoritma berangkat dari himpunan solusi (populasi) yang dihasilkan secara acak. Masing-masing individu dalam populasi disebut *chromosom* yang merupakan representasi dari suatu solusi (Berlianti dan Arifin, 2010). cara kerja Algoritma Genetika pada dasarnya yaitu membangkitkan populasi secara acak dimana populasi terdiri dari individu-individu yang dibangkitkan secara acak.

Selanjutnya masing-masing individu akan dievaluasi berdasarkan nilai *fitness* dari masing-masing individu acak tadi, kemudian dipilih individu dengan nilai *fitness* terbaik

(semakin tinggi nilai *fitness* maka semakin besar peluang individu tersebut terpilih menjadi kromosom induk). Individu terpilih ini (kromosom induk) selanjutnya akan saling ditukar untuk membentuk populasi baru. Pertukaran individu terdiri dari dua jenis mekanisme, yaitu kawin silang (*cross over*) dan mutasi. cara kerja Algoritma Genetika pada dasarnya yaitu membangkitkan populasi secara acak dimana populasi terdiri dari individu-individu yang dibangkitkan secara acak.

Penelitian ini akan mengembangkan metode algoritma *Cross Entropy-Genetic Algorithm* (CEGA) pada kasus penjadwalan *flow shop* dengan tujuan untuk mendapatkan nilai *makespan* yang minimal. Dengan harapan dapat meminimasi waktu proses sehingga akan meningkatkan efisiensi utilitas produksi.

Pemilihan metode CEGA ini dipilih karena berdasarkan hasil penelitian yang telah dilakukan oleh Budiman (2010) dengan menggunakan metode CEGA pada kasus *job shop* yang bertujuan meminimasi *makespan* mampu mendapatkan hasil performansi yang lebih baik berdasarkan nilai *makespan* dan waktu komputasi jika dibandingkan dengan algoritma lain seperti *genetic algorithm-simulated annealing*, dan *hybrid tabu search*. Berdasarkan hasil tersebut maka pada penelitian ini akan mengembangkan metode CEGA dengan pengembangan untuk diterapkan pada kasus *flow shop scheduling*.

METODE PENELITIAN

Lokasi Penelitian

Penelitian ini dilakukan di PT.X yang berlokasi di Krian Kabupaten Sidoarjo Jawa Timur selama 6 bulan sejak September 2014-Februari 2015.

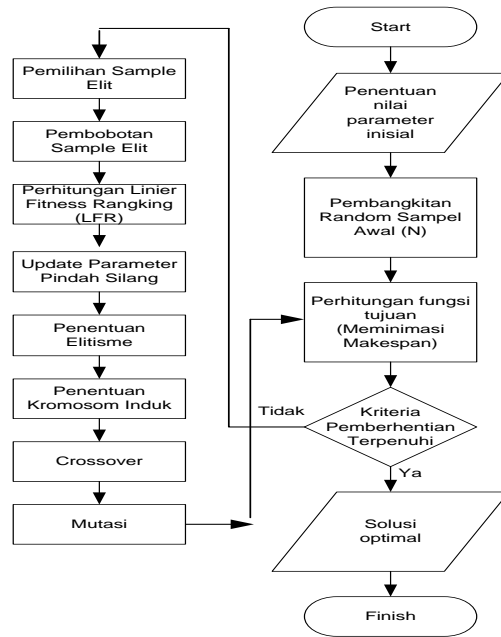
Pengumpulan Data

Pengumpulan data dilakukan dengan dua teknik yaitu studi pustakan dan studi lapangan yang meliputi observasi, wawancara, dan data historis perusahaan.

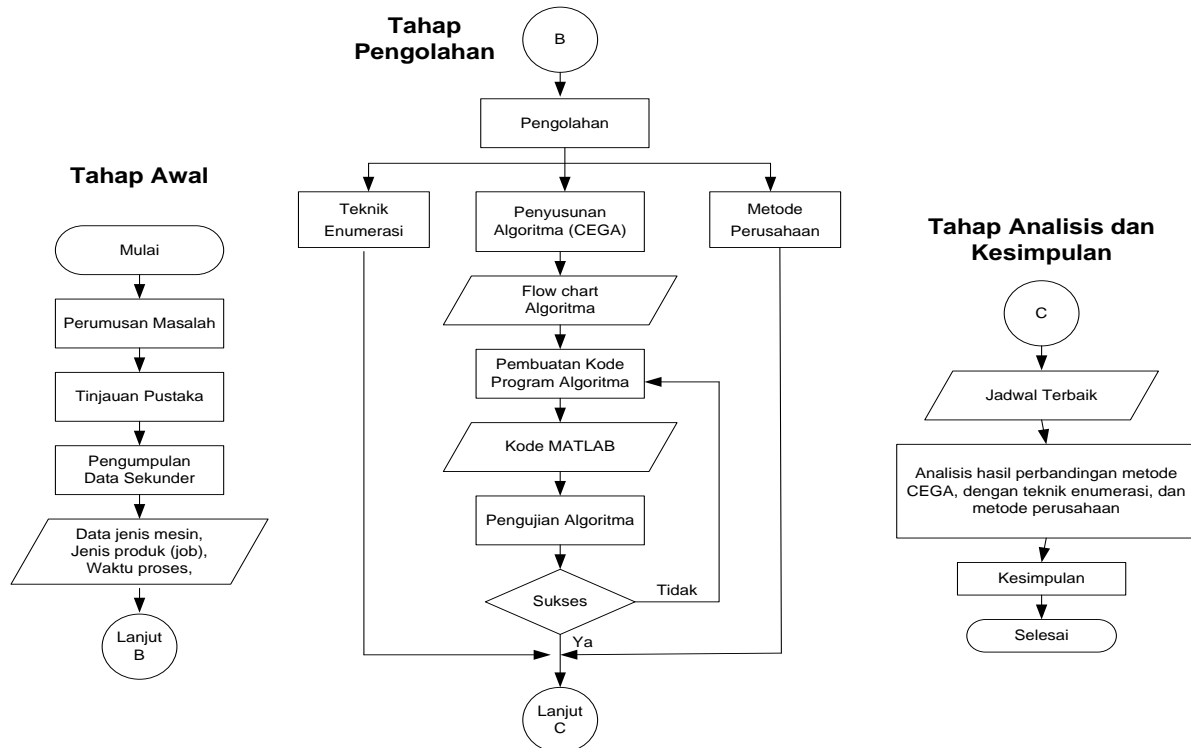
Tahapan Penelitian

Pengembangan metode penelitian pada algoritma *Cross Entropy* ini nantinya akan digabungkan dengan Algoritma Genetika. Tujuan pengembangan ini adalah memperluas pencarian solusi pada sampel elit *Cross Entropy* (CE) ketika berada di lokal optimal dengan menambahkan mekanisme pindah silang dan mutasi pada Algoritma Genetika (AG), hal tersebut berfungsi untuk menghindari kemungkinan pencarian solusi terjebak di area lokal optimal karena mekanisme GA dapat menghasilkan kromosom baru yang memiliki sifat tetap menjaga nature kromosom pada populasi awalnya (Widodo, Santoso, dan Siswanto, 2014).

Berikut ini tahapan metode Algoritma CEGA yang diusulkan di kutip dari penelitian yang pernah dilakukan oleh (Nurkhalida dan Santosa, 2012) dapat ditunjukkan pada Gambar 1. Sedangkan langkah-langkah penelitian secara keseluruhan ditunjukkan pada Gambar 2. Penelitian ini juga akan melakukan perbandingan antara perhitungan dengan teknik enumerasi, hasil komputasi Algoritma CEGA, dan metode perusahaan.



Gambar 1. Tahapan Metode Pengembangan Algoritma CEGA



Gambar 2. Tahapan metode penelitian

HASIL DAN PEMBAHASAN

Perhitungan Teknik *Enumerasi*

Untuk menghitung nilai *makespan* pada perhitungan teknik enumerasi ini digunakan secara manual dengan bantuan *Microsoft excel*. Data inputan yang dipakai yaitu waktu standart mesin pada pengerjaan tiap pekerjaan. Berikut ini waktu standart pengerjaan mesin tiap pekerjaan akan disajikan pada Tabel 1.

Tabel 1. Waktu Standart Pengerjaan Mesin Tiap Pekerjaan Dalam Satuan Detik

Mesin	WAKTU BAKU MESIN (detik)				
	Pekerjaan	Pekerjaan	Pekerjaan	Pekerjaan	Pekerjaan
	1	2	3	4	5
Cuting Machine (M1)	614	812	977	1012	1421
Welding (M2)	893	0	0	987	1210
Turning Machine (M3)	1118	1298	1184	1322	1586
Hobing Machine (M4)	1321	1534	1098	2158	2608

Sumber: Data dari Perusahaan

Keterangan:

- Pekerjaan 1 = Roda gigi Ø 60 mm, Pekerjaan 2 = Roda gigi tajam Ø 85 mm
- Pekerjaan 3 = Roda gigi tumpul Ø 85 mm, Pekerjaan 4 = Roda gigi Ø 120 mm
- Pekerjaan 5 = Roda gigi Ø 300 mm

Penyelesaian yang akan dilakukan dengan teknik enumerasi ini untuk mendapatkan urutan pengerjaan pekerjaan yang optimal sehingga akan didapatkan nilai fungsi tujuan (nilai *makespan*) yang minimum. Adapun menurut Widodo, Santoso dan Siswanto (2014) urutan proses penyelesaian permasalahan dengan teknik enumerasi ini adalah sebagai berikut:

- a. Tahap 1: Pembangkitan secara acak urutan pekerjaan
 Pada tahap ini dilakukan pembangkitan secara acak urutan pekerjaan yang mungkin akan terjadi.
- b. Tahap 2: Perhitungan nilai *makespan*
 Pada tahap ini setiap urutan jadwal (pekerjaan) yang dibangkitkan pada tahap ini akan dilakukan perhitungan *makespan*.
- c. Tahap 3: Pemilihan solusi terbaik
 Pada tahap ini semua urutan penjadwalan pekerjaan yang telah dihitung nilai *makespan* pada tahap sebelumnya kemudian akan dipilih solusi terbaik yaitu berdasarkan nilai *makespan* terkecil.

Dari data waktu standart pengerjaan Mesin tiap pekerjaan kemudian dilakukan perhitungan dengan teknik enumerasi. Pada penelitian ini dimana terdapat 5 pekerjaan yang akan di jadwalkan maka total kemungkinan penjadwalan urutan pekerjaan yaitu $5! = 2 \times 3 \times 4 \times 5 = 120$ total kemungkinan jadwal urutan pekerjaan. Akan tetapi pada hasil

perhitungan *makespan* untuk teknik enumerasi ini hanya akan ditunjukkan 16 jadwal pengerjaan pekerjaan secara acak dan hasil untuk perhitungan ditunjukkan pada Tabel 2. Dari hasil perhitungan dengan teknik enumerasi tersebut menunjukkan bahwa terdapat 4 alternatif penjadwalan pekerjaan yang memiliki nilai *makespan* minimum sebesar 10829 detik yaitu pada urutan pekerjaan 2-1-3-4-5, pekerjaan 2-1-4-5-3, pekerjaan 2-1-4-3-5, dan pekerjaan 2-3-1-4-5

Tabel 2. Nilai *Makespan* Berdasarkan Urutan Pekerjaan Secara Acak

URUTAN PEKERJAAN	MAKESPAN (detik)
2-5-3-1-4	12214
4-3-2-1-5	13386
2-1-3-4-5	10829
4-2-1-3-5	13386
1-2-3-4-5	11344
3-4-1-2-5	13265
5-3-2-4-1	12936
2-1-4-5-3	10829
2-1-4-3-5	10829
2-3-1-4-5	10829
1-4-2-3-5	11345
1-4-3-2-5	11345
2-3-4-1-5	11197
3-1-4-5-2	11224
3-1-2-4-5	11223
3-2-4-1-5	11197

Perhitungan Algoritma CEGA

Pada proses perhitungan algoritma CEGA ini akan diberikan untuk perhitungan satu iterasi dalam menyelesaikan contoh kasus yang sama yang telah dilakukan sebelumnya pada teknik enumerasi. Adapun langkah-langkah untuk melakukan perhitungan algoritma CEGA sebagai berikut:

Langkah 1: Inisialisasi Parameter

Untuk langkah inisialisasi parameter algoritma CEGA adapun parameter-parameter yang dibutuhkan untuk melakukan perhitungan CEGA sebagai berikut:

- Jumlah sampel yang dibangkitkan (N) = 7
- Parameter kejarangan (ρ) = 0.03
- Koefisien penghalusan (α) = 0.5
- Parameter pindah silang (P_{ps}) = 1
- Parameter pemberhentian (β) = 0.0001

Langkah 2: Pembangkitan Sample

Pada langkah pembangkitan sample ini akan dibangkitkan secara acak yang nantinya akan mewakili prioritas urutan pekerjaan dari seluruh operasi pengerjaan pekerjaan. Dalam penenilitian ini akan dilakukan pembangkitan bilangan acak yang membentuk 7 sampel sebagai berikut:

$$\begin{matrix} Z1 = 1-2-3-4-5 & Z3 = 4-2-1-3-5 & Z5 = 3-4-1-2-5 & Z7 = 2-5-3-1-4 \\ Z2 = 2-1-3-4-5 & Z4 = 4-3-2-1-5 & Z6 = 5-3-2-4-1 \end{matrix}$$

Pembangkitan sample secara random ini berupa urutan pengerjaan pekerjaan, dimana sampel Z1 urutan pengerjaan pekerjaan nya yaitu pekerjaan 1, pekerjaan 2, pekerjaan 3, pekerjaan 4, dan pekerjaan 5, sampel Z2 urutan pengerjaan pekerjaan nya yaitu pekerjaan 2, pekerjaan 1, pekerjaan 3, pekerjaan 4, dan pekerjaan 5, dan begitu juga untuk urutan pengerjaan pekerjaan untuk Z3 sampai Z7.

Langkah 3: Perhitungan Fungsi Tujuan yaitu Makespan

Pada langkah perhitungan fungsi tujuan akan dihitung berdasarkan nilai makespan, berikut ini diberikan contoh perhitungan satu sampel yaitu pada Z1 dan untuk sampel lain dihitung dengan langkah yang sama. Sampel yang digunakan adalah sampel pertama Z1 1-2-3-4-5. Urutan pengerjaan pekerjaan pertama dari sampel 1-2-3-4-5 adalah pekerjaan ke 1.

Untuk menentukan nilai waktu mulai dan waktu selesai pada proses perhitungan fungsi tujuan menggunakan syarat sebagai berikut:

- a. Jika operasi tersebut tidak memiliki operasi prasyarat (pekerjaan) dan operasi pendahulu (mesin), maka letakkan operasi dengan waktu mulai = 0.
- b. Jika tidak terdapat operasi prasyarat (pekerjaan) namun terdapat operasi pendahulu (mesin), maka waktu mulai operasi = waktu selesai operasi pendahulu.
- c. Jika tidak terdapat operasi pendahulu (mesin) namun terdapat operasi prasyarat (pekerjaan), maka waktu mulai operasi = waktu selesai operasi prasyarat (pekerjaan).
- d. Jika terdapat operasi prasyarat (pekerjaan) dan operasi pendahulu (mesin), maka waktu mulai operasi = waktu selesai terlama diantara operasi prasyarat (pekerjaan) dan operasi pendahulu.

Dimana pada operasi prasyarat memiliki hubungan dalam satu pekerjaan, sedangkan operasi pendahulu memiliki hubungan dalam operasi mesin. Berikut ini hasil perhitungan makespan pada penjadwalan pekerjaan untuk sampel Z1 1-2-3-4-5 didapatkan total makespan sebesar 11344 detik. Untuk hasil perhitungannya dapat dilihat pada Table 3.

Tabel 3. Nilai Fungsi Tujuan Pada Pekerjaan Sampel Z1.

URUTAN PEKERJAAN	WAKTU STANDART MESIN (Detik)				MULAI (Detik)				SELESAI (Detik)			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
1	614	893	1118	1321	0	614	1507	2625	614	1507	2625	3946
2	812	0	1298	1534	614	1507	2625	3946	1426	1507	3923	5480
3	977	0	1184	1098	1426	2403	3923	5480	2403	2403	5107	6578
4	1012	987	1322	2158	2403	3415	5107	6578	3415	4402	6429	8736
5	1421	1210	1586	2608	3415	4836	6429	8736	4836	6046	8015	11344

Tabel 4. Hasil Perhitungan *Makespan* Untuk 7 Sample

No	Sample	Urutan Pekerjaan	Makespan (Detik)
1	Z1	1-2-3-4-5	11344
2	Z2	2-1-3-4-5	10829
3	Z3	4-2-1-3-5	13386
4	Z4	4-3-2-1-5	13386
5	Z5	3-4-1-2-5	13265
6	Z6	5-3-2-4-1	12936
7	Z7	2-5-3-1-4	12214

Rumus yang digunakan untuk menghitung nilai *Makespan* (Z) adalah

$$Z = \max_{1 \leq j \leq n} (F_j) \quad (1)$$

Sehingga didapatkan maksimum *flow time* (F_j) yaitu 11344 detik yang artinya bahwa nilai *makespan* (Z) sebesar 11344 detik. Untuk hasil perhitungan *makespan* didapatkan bahwa nilai *makespan* minimum terletak pada sample Z2 dengan nilai *makespan* sebesar 10829 detik, untuk nilai *makespan* pada tujuh sampel dapat dilihat pada Tabel 4.

Tabel 5. Hasil Pengurutan Nilai *Makespan* Untuk 7 Sample

No Urut	Sample	Urutan Pekerjaan	Makespan (Detik)
1	Z2	2-1-3-4-5	10829
2	Z1	1-2-3-4-5	11344
3	Z7	2-5-3-1-4	12214
4	Z6	5-3-2-4-1	12936
5	Z5	3-4-1-2-5	13265
6	Z3	4-2-1-3-5	13386
7	Z4	4-3-2-1-5	13386

Langkah 4: Penentuan Sampel Elit

Penentuan nilai sampel elit ini didasarkan pada langkah inisialisasi parameter sebelumnya telah dipilih nilai ρ sebesar 0.03, maka rumus jumlah sampel elit yaitu $N \times \rho = 7 \times 0.03 = 0.21 \approx 1$ (Nurkhalida, dan Santosa, 2012). Dari perhitungan nilai sampel elit maka akan diambil satu sampel teratas yang memiliki makespan terkecil sebagai sampel elit yaitu terdapat pada nilaiurut ke Z1 atau pada sampel ke-2 (Z2) dengan urutan prioritas pekerjaan yaitu 2-1-3-4-5. Untuk penentuan sampel elit tersebut nilai *makespan* dari ke 7 sampel kemudian di urutkan dari yang terkecil hingga terbesar. Untuk hasil pengurutan nilai *makespan* untuk 7 sample dapat dilihat pada Tabel 5. Hasil dari pengurutan nilai makespan ini nantinya akan dipakai sampai langkah tahapan terakhir.

Langkah 5: Pembobotan Sampel Elit

Nilai pembobotan sampel elit ini didasarkan dari perhitunngan penentuan sampel elit pada langkah sebelumnya yang didapatkan hasil bahwa hanya terdapat 1 sampel elit, maka bobot sampel elit tersebut = 1 nilai tersebut didapatkan dari evaluasi terhadap nilai terbaik pada iterasi sebelumnya (Nurkhalida, dan Santosa, 2012).

Langkah 6: Perhitungan Linier Fitness Ranging (LFR)

Tahap perhitungan nilai LFR ini digunakan untuk pemilihan induk pada proses pindah silang. LFR diperoleh melalui rumus (Nurkhalida dan Santosa, 2012):

$$LFR(I(N-i+1)) = F_{max} - (F_{max} - F_{min}) * ((i-1)/(N-1)) \tag{2}$$

dengan $F_{max} = 1/Z(1) = 1/10829 = 0,00009234$ dan $F_{min} = 1/Z(N) = 1/Z(7) = 1/13386 = 0,00007470$, dari hasil *makespan* yang sudah di urutkan maka LFR untuk ke 7 sampel adalah:

$$Z1 \Rightarrow Z2 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((1-1)/(7-1))) = 0,0000923446$$

$$Z2 \Rightarrow Z1 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((2-1)/(7-1))) = 0,0000898247$$

$$Z3 \Rightarrow Z7 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((3-1)/(7-1))) = 0,0000873047$$

$$Z4 \Rightarrow Z6 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((4-1)/(7-1))) = 0,0000847848$$

$$Z5 \Rightarrow Z5 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((5-1)/(7-1))) = 0,0000822648$$

$$Z6 \Rightarrow Z3 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((6-1)/(7-1))) = 0,0000797448$$

$$Z7 \Rightarrow Z4 \rightarrow LFR = 0,00009234 - ((0,00009234 - 0,00007470) * ((7-1)/(7-1))) = 0,0000772249$$

Langkah 7: Update Parameter Pindah Silang

Pada langkah *Update* parameter pindah silang ini digunakan untuk mendapatkan nilai parameter yang *update* untuk evaluasi kriteria pemberhentian. Semakin besar nilai parameter pindah silang maka semakin banyak jumlah sampel yang akan mengalami pindah silang. Maka menurut Nurkhalida dan Santosa (2012) rumus *update* parameter pindah silang adalah

$$P_{ps(i)} = (1-\alpha) * u + (P_{ps(i+1)}) * \alpha \tag{3}$$

Dengan u adalah

$$u = \frac{\overline{Z_e}}{2 * Z_{best}} \tag{4}$$

Dimana Z_e = objektif pada sampel elite dan Z_{best} = objektif terbaik pada tiap iterasi. Maka nilai parameter pindah silang (P_{ps}) adalah

$$u = \frac{\overline{Z_e}}{2 * Z_{best}} = \frac{10829}{2 * 10829} = 0,5$$

$$\begin{aligned} P_{ps(i)} &= (1-\alpha)*u + (P_{ps(i+1)})*\alpha \\ &= (1-0,5)*0,5 + (1)*0,5 \\ &= 0,75 \end{aligned}$$

Langkah 8: Elitisme

Langkah *elitisme* ini bertujuan untuk menyimpan sampel dengan nilai fungsi tujuan terbaik pada setiap iterasi. Oleh karenanya untuk menjaga agar individu bernilai *fitness* tertinggi tersebut tidak hilang selama evolusi, sample ini nantinya akan muncul kembali pada populasi sampel di iterasi berikutnya (Nurkhalida dan Santosa, 2012). Sedangkan jumlah sampel yang di elitism ini sebanyak satu sampel, karena dari hasil perhitungan sampel elit didapatkan nilai 1. Sehingga sampel yang di *elitisme* adalah $Z1 = 2-1-3-4-5$.

Langkah 9: Penentuan Pemilihan Induk Pindah Silang

Pemilihan induk pindah silang dilakukan dengan memilih dua buah kromosom sebagai induk yang akan dipindah silangkan dilakukan secara proporsional sesuai dengan dengan nilai *fitness*-nya. Dalam penelitian ini untuk melakukan pindah silang menggunakan mekanisme *roulette wheel*, yaitu dengan pemilihan induk 1 dari sampel elit dan induk 2 dari sampel keseluruhan.

1. Menentukan Pemilihan induk 1

Untuk menentukan pemilihan induk 1 ini akan di ambil dari nilai sampel elit yang memiliki nilai fungsi tujuan terbaik (nilai makespan minimum) dari mekanisme elitisme sehingga didapatkan pada sampel $Z1 = 2-1-3-4-5$.

2. Menentukan Pemilihan induk 2

Untuk menentukan pemilihan induk 2 ini akan dipilih berdasarkan nilai evaluasi dari sampel keseluruhan. Pada proses ini akan menggunakan nilai LFR dari sampel yang sedang dievaluasi. Apabila nilai perbandingan antara kumulatif LFR dan total LFR lebih besar dari nilai acak yang dibangkitkan, maka sampel tersebut menjadi induk 2 (Nurkhalida dan Santosa, 2012).

$$\begin{aligned} \text{Total LFR} &= 0,0000923446 + 0,0000898247 + 0,0000873047 + 0,0000847848 + \\ &0,0000822648 + 0,0000797448 + 0,0000772249 = 0,0005934933 \end{aligned}$$

Nilai hasil dari pembangkitan bilangan acak didapatkan nilai sebesar 0,6580. Kemudian hasil kumulatif LFR total dibandingkan nilai yang dibangkitkan secara acak adalah sebagai berikut:

$$Z1 = 0,0000923446 / 0,0005934933 = 0,156 < 0,6580$$

$$Z2 = [0,0000923446 + 0,0000898247] / 0,0005934933 = 0,307 < 0,6580$$

$$Z3 = [0,0000923446 + 0,0000898247 + 0,0000873047] / 0,0005934933 = 0,454 < 0,6580$$

$$Z4 = [0,0000923446 + 0,0000898247 + 0,0000873047 + 0,0000847848] / 0,0005934933 = 0,597 < 0,6580$$

$$Z5 = [0,0000923446 + 0,0000898247 + 0,0000873047 + 0,0000847848 + 0,0000822648] / 0,0005934933 = 0,736 > 0,6580$$

$$Z6 = [0,0000923446 + 0,0000898247 + 0,0000873047 + 0,0000847848 + 0,0000822648 + 0,0000797448] / 0,0005934933 = 0,870 > 0,6580$$

$$Z7 = [0,0000923446 + 0,0000898247 + 0,0000873047 + 0,0000847848 + 0,0000822648 + 0,0000797448 + 0,0000772249] / 0,0005934933 = 1 > 0,6580$$

Berdasarkan hasil perhitungan diatas, diketahui sampel Z1 dijadikan induk 1 didapatkan dari nilai sampel elit yang memiliki nilai fungsi tujuan terbaik (nilai makespan minimum) dari mekanisme elitism, sedangkan sampel Z5, Z6 dan Z7 ditetapkan sebagai induk 2 didapatkan dari nilai perbandingan antara kumulatif LFR dan total LFR lebih besar dari nilai acak.

Langkah 10: Cross Over (Pindah Silang)

Dari hasil langkah pemilihan induk pindah silang diperoleh bahwa Z1 ditetapkan sebagai induk 1 serta Z5, Z6, dan Z7, sebagai induk 2 selanjutnya akan dilakukan teknik *Cross over*. *Cross over* (Pindah Silang) merupakan tahapan dari algoritma genetik yang akan menyilangkan 2 induk untuk membentuk kromosom baru yang akan menghasilkan individu baru yang lebih baik. Metode *cross over* yang digunakan dalam penelitian ini adalah *2-point order cross over*.

Dalam teknik *cross over* ini untuk mengetahui sampel mana yang akan mengalami *cross over* maka dilakukan penbangkitan bilangan acak. Jika dalam pembangkitan bilangan acak diperoleh bilangan random (R) lebih kecil dari parameter pindah silang (P_{ps}) atau R < P_{ps}, maka akan dilakukan pindah silang terhadap kedua kromosom induk tersebut. Namun jika pembangkitan bilangan R lebih besar dari parameter pindah silang (P_{ps}) atau R > P_{ps}, maka tidak terjadi pindah silang terhadap kedua kromosom induk. Hasil dari pembangkitan bilangan R dapat dilihat pada Tabel 6.

Pada Tabel 6 sample Z2, Z3, Z4, Z5, Z6, dan Z7 didapatkan hasil bilangan acak yang dibangkitkan nilainya lebih kecil dari P_{ps} (0,75) sehingga nantinya akan dilakukan pindah silang. Dalam melakukan pindah silang ini induk yang dipakai yaitu sample Z1 dan hasil pemilihan induk 2 dipilih sample Z5, Z6, dan Z7 (berdasarkan hasil langkah 9).

Tabel 6. Pembangkitan Bilangan R Penentuan Cross Over

No	Sample	Urutan Pekerjaan	R	P _{ps}	Kesimpulan
Z1	Z2	2-1-3-4-5	0.8643	0.75	Tanpa Cros over
Z2	Z1	1-2-3-4-5	0.6231	0.75	Cros over
Z3	Z7	2-5-3-1-4	0.5129	0.75	Cros over
Z4	Z6	5-3-2-4-1	0.4885	0.75	Cros over
Z5	Z5	3-4-1-2-5	0.6643	0.75	Cros over
Z6	Z3	4-2-1-3-5	0.6883	0.75	Cros over
Z7	Z4	4-3-2-1-5	0.3865	0.75	Cros over

Untuk menentukan bagian dari sampel induk yang akan ditukar, maka dilakukan mekanisme dengan membangkitkan dua bilangan acak, hal ini dilakukan untuk proses pindah silang pada *ceil* atau gen induk yang akan mengalami proses *cross over*. Pada proses pindah silang penentuan *ceil* yang mengalami perubahan tersebut ini diperoleh bilangan acak 0.1325 dan 0.5922 (Tabel 7). Nilai acak tersebut kemudian dikonversi menjadi nilai bulat, dan digunakan sebagai pembatas bagian sampel yang akan di pindah antar induk.

$$r_i = \text{ceil}(\text{acak} * n) \tag{5}$$

$$r_1 = \text{ceil}(0.1325 * 5) = 1 \text{ maka } p_1 = 1$$

$$r_2 = \text{ceil}(0.5922 * 5) = 3 \text{ maka } p_2 = 2$$

Induk 1 = 2 | **1 3** | 4 5
 Induk 2 = 3 | **4 1** | 2 5

Selanjutnya angka yang berada didalam kurung tersebut akan ditukar antar induk, sehingga menjadi seperti berikut:

Calon anak 1 = ... **4 1** ...
 Calon anak 2 = ... **1 3** ...

Untuk mengisi titik-titik pada calon anak 1, maka diperiksa satu per satu operasi pada induk1. Jika terdapat operasi pada induk 1 yang belum tercantum pada calon anak 1, maka operasi tersebut akan mengisi titik-titik. Hal ini dilakukan secara berurutan. Sehingga sampel anakan hasil pindah silang menjadi:

Anak 1 = 2-4-1-5-3 → menggantikan Z2 lama
 Anak 2 = 2-1-3-5-4 → menggantikan Z3 lama

Tabel 7. Proses *Cross over* Pada *Ceil*

Induk Yang di Cross Over	Urutan Job	Pembangkit an bil Random	$r_i = \text{ceil}(\text{random} * n)$ dan $n=5$	Ceil yang di cross over	Hasil Cros over	Keterangan	
1	2	3	5	6	7	8	
Z1 VS Z5	2-1-3-4-5	r1	0.1325	0.6625 = 1	2-1-3-4-5	2-4-1-5-3	Menjadi sampel Z2
	3-4-1-2-5	r2	0.5922	2.9610 = 3	3-4-1-2-5	2-1-3-5-4	Menjadi sampel Z3
Z1 VS Z6	2-1-3-4-5	r1	0.1896	0.9480 = 1	2-1-3-4-5	3-2-4-5-1	Menjadi sampel Z4
	4-2-1-3-5	r2	0.3859	1.9295 = 2	4-2-1-3-5	4-1-3-5-2	Menjadi sampel Z5
Z1 VS Z7	2-1-3-4-5	r1	0.6021	3.0105 = 3	2-1-3-4-5	2-3-5-1-4	Menjadi sampel Z6
	4-3-2-1-5	r2	0.7791	3.8955 = 4	4-3-2-1-5	3-2-5-4-1	Menjadi sampel Z7

Kemudian ulangi dengan mengevaluasi urutan sampel ke 4, 5, 6 dan 7. Dengan perhitungan yang sama dengan awal, diketahui induk1 = Z1 dan induk2 = Z5, Z6 dan Z7. Dengan acak lebih kecil dari P_{ps} (0.75), maka terjadi pindah silang. Induk1 dan induk2 digunakan untuk mengganti urutan ke- 4, ke- 5, ke- 6 dan ke-7 untuk hasil selengkapnya dapat dilihat pada Tabel 8.

Tabel 8. Hasil *Cross Over* (Pindah Silang)

No	Sample	Urutan Job	Random (R)	P_{ps}	Kesimpulan	Hasil Cros over	Keterangan
Z1	Z2	2-1-3-4-5	0.8643	0.75	Tanpa Cros over	2 - 1 - 3 - 4 - 5	Diambil dari Kromosom Induk Z1
Z2	Z1	1-2-3-4-5	0.6231	0.75	Cros over	2 - 4 - 1 - 5 - 3	Hasil cross over induk Z1 VS Z5
Z3	Z7	2-5-3-1-4	0.5129	0.75	Cros over	2 - 1 - 3 - 5 - 4	Hasil cross over induk Z1 VS Z5
Z4	Z6	5-3-2-4-1	0.4885	0.75	Cros over	3 - 2 - 4 - 5 - 1	Hasil cross over induk Z1 VS Z6
Z5	Z5	3-4-1-2-5	0.6643	0.75	Cros over	4 - 1 - 3 - 5 - 2	Hasil cross over induk Z1 VS Z6
Z6	Z3	4-2-1-3-5	0.6883	0.75	Cros over	2 - 3 - 5 - 1 - 4	Hasil cross over induk Z1 VS Z7
Z7	Z4	4-3-2-1-5	0.3865	0.75	Cros over	3 - 2 - 5 - 4 - 1	Hasil cross over induk Z1 VS Z7

Langkah 11: Mutasi

Langkah selanjutnya yaitu melakukan tahap mutasi. Mutasi ini merupakan bagian dari algoritma genetika untuk memunculkan individu baru yang berbeda dengan individu yang sudah ada. Salah satu bagian penting dalam tahap mutasi ini yaitu menentukan nilai *probabilitas* mutasi (P_m) yang nantinya akan menentukan kromosom mana yang akan mengalami perubahan gen, semakin besar nilai peluang mutasi maka semakin banyak kromosom dalam populasi yang akan mengalami mutasi (Santosa dan Willy, 2011). Proses mutasi akan dipilih secara acak dan *gen* pada site tersebut akan diubah nilainya. Angka acak akan dibangkitkan dengan batasan 0 sampai 1. Jika angka acak (R) tersebut lebih kecil dari parameter mutasi (P_m) maka digit gen akan diganti, dan jika angka acak (R) tersebut lebih besar dari parameter mutasi (P_m) maka digit gen tidak akan diganti.

Menurut Hanka dan Santosa (2013) nilai parameter mutasi ditentukan dengan rumus sebagai berikut:

$$P_m = \frac{P_{ps}}{2} = \frac{0,75}{2} = 0,375 \tag{6}$$

Setelah didapatkan nilai parameter mutasi (P_m) sebesar 0.375, dan N sebesar 7. Maka jumlah sampel kromosom yang akan mengalami mutasi adalah:

$N_a = P_m \times N = 0,375 \times 7 = 2,625 \approx 3$; sehingga ada 3 sampel kromosom yang akan mengalami proses mutasi. Untuk mengetahui kromosom mana yang akan mengalami mutasi maka akan dibangkitkan bilangan acak. Berikut ini hasil pembangkitan bilangan acak pada kromosom yang akan mengalami proses mutasi ditunjukkan pada Tabel 9.

Tabel 9. Penentuan Kromosom Yang Akan Dimutasi

Sample	Hasil Crossover	Random mutasi	Probabilitas Mutasi (Pm)	Keterangan
Z1	2 - 1 - 3 - 4 - 5	Z1 Tetap diambil dari pemilihan induk 1		
Z2	2 - 4 - 1 - 5 - 3	0.2338	0,375	$R < P_m$ maka dimutasi
Z3	2 - 1 - 3 - 5 - 4	0.5876	0,375	$R > P_m$ maka tidak dimutasi
Z4	3 - 2 - 4 - 5 - 1	0.8105	0,375	$R > P_m$ maka tidak dimutasi
Z5	4 - 1 - 3 - 5 - 2	0.1884	0,375	$R < P_m$ maka dimutasi
Z6	2 - 3 - 5 - 1 - 4	0.7491	0,375	$R > P_m$ maka tidak dimutasi
Z7	3 - 2 - 5 - 4 - 1	0.2464	0,375	$R < P_m$ maka dimutasi

Dari Table 9 maka kromosom yang akan mengalami mutasi adalah Z2, Z5, dan Z7. Maka tahap selanjutnya yaitu melakukan mutasi pada gen ke 3 kromosom tersebut. Berikut akan diberikan contoh proses mutasi pada sample kromosom Z2:

$$\text{rand } 1 = 0.2022; \text{ rand } 2 = 0.7943; n = 5$$

$$I, J = \text{ceil}(n * r_{ij}) \quad (7)$$

$$I = \text{ceil}(5 * 0.2022) = 1.011 \approx 1$$

$$J = \text{ceil}(5 * 0.7943) = 3.972 \approx 4$$

Jika nilai:

- k = 1 : maka dilakukan pembalikan (*flip mutation*)
- k = 2 : maka dilakukan penukaran (*swap mutation*)
- k = 3 : maka dilakukan penggeseran (*slide mutation*)

$$\text{rand } K = 0.3612$$

$$k = \text{ceil}(\text{rand } K * 3) \quad (8)$$

$$k = \text{ceil}(0.3612 * 3) = 1,08 \approx 1 \text{ berarti dilakukan pembalikan}$$

Z2 = 2 – 4 – 1 – 5 – 3 dilakukan pembalikan, sehingga proses mutasi menjadi
Z2 = 2 – 5 – 1 – 4 – 3

Untuk mengetahui proses mutasi yang terjadi pada Z5 dan Z7, dilakukan langkah yang sama seperti pada proses mutasi Z2. Sehingga didapatkan proses mutasi sampel Z5 dan Z7 berubah menjadi seperti pada Table 10.

Tabel 10. Hasil Proses Mutasi

Sample	Hasil Cros over	Random mutasi	Probabilitas Mutasi (Pm)	Keterangan	Random (r1,r2)	Nilai I,J dengan n=5	Random K	Nilai K dengan k=3	Proses Mutasi	Hasil mutasi
1	2	3	4	5	6	7=(Kolom 6*n)	8	9=(Kolom 8*k)	10	11
Z1	2-1-3-4-5	Z1 Tetap diambil dari pemilihan induk 1								2-1-3-4-5
Z2	2-4-1-5-3	0.2338	0,375	R < Pm maka dilakukan mutasi	r1 0.2022 r2 0.7943	I 1.011 = 1 J 3.972 = 4	0.3612	1.08 = 1	Flip mutation (membalik)	2-5-1-4-3
Z3	2-1-3-5-4	0.5876	0,375	R > Pm maka tidak dilakukan						2-1-3-5-4
Z4	3-2-4-5-1	0.8105	0,375	R > Pm maka tidak dilakukan						3-2-4-5-1
Z5	4-1-3-5-2	0.1884	0,375	R < Pm maka dilakukan mutasi	r1 0.1863 r2 0.9571	I 0.932 = 1 J 4.786 = 5	0.8583	2.57 = 3	Slide mutation (menggeser)	4-5-1-3-2
Z6	2-3-5-1-4	0.7491	0,375	R > Pm maka tidak dilakukan						2-3-5-1-4
Z7	3-2-5-4-1	0.2464	0,375	R < Pm maka dilakukan mutasi	r1 0.2144 r2 0.9351	I 1.072 = 1 J 4.676 = 5	0.6643	1.99 = 2	Swap mutation (menukar)	3-1-5-4-2

Langkah 12: Menghitung Nilai Fungsi Tujuan (*Makespan*) Dari Populasi Baru

Pada langkah akhir proses mutasi maka akan diperoleh sample baru berupa urutan pekerjaan yang baru, dan nantinya sample baru tersebut akan di hitung nilai *makespannya*. Berikut ini hasil perhitungan nilai fungsi tujuan (*makespan*) dari sample baru terdapat dalam Tabel 11.

Tabel 11. Hasil Perhitungan *Makespan* Populasi Baru

Sample	Urutan Pekerjaan Baru	Nilai Makespan (Detik)
Z1	2 - 1 - 3 - 4 - 5	10829
Z2	2 - 5 - 1 - 4 - 3	12214
Z3	2 - 1 - 3 - 5 - 4	11386
Z4	3 - 2 - 4 - 5 - 1	11197
Z5	4 - 5 - 1 - 3 - 2	12040
Z6	2 - 3 - 5 - 1 - 4	12093
Z7	3 - 1 - 5 - 4 - 2	12108

Untuk menguji hasil algoritma CEGA maka berikut ini ditampilkan hasil keluran penjadwalan algoritma CEGA pada kasus *flow shop* dengan menggunakan program MATLAB.

```

New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> [JadwalOpt, Makespan, it, T]=flowshopscheduling(7,0.03,0.5,5,4,1,t)

JadwalOpt =

     2     1     3     4     5

Makespan =

    10829

it =

     14

T =

    0.0936
    
```

Gambar 3. Keluaran Solusi Optimal *Coding* algoritma CEGA Dengan MATLAB

Perhitungan Metode Yang Ada Perusahaan

Untuk menjalankan aktivitas produksi selama ini, perusahaan menggunakan model penjadwalan yang cenderung berubah-ubah, jadi tidak memiliki model penjadwalan yang pasti. Maka untuk memilih metode penjadwalan perusahaan akan dipilih model pengerjaan pekerjaan yang paling sering dilakukan oleh perusahaan. Sedangkan penjadwalan pengerjaan pekerjaan yang dilakukan perusahaan yaitu 4-5-1-3-2 dengan total waktu makespan sebesar 12040 detik. Untuk mengetahui hasil perhitungan *makespan* dapat dilihat pada Tabel 12.

Tabel 12. Nilai *Makespan* Perusahaan

URUTAN JOB	WAKTU STANDART MESIN (Detik)				MULAI (Detik)				SELESAI (Detik)			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
4	1012	987	1322	2158	0	1012	1999	3321	1012	1999	3321	5479
5	1421	1210	1586	2608	1012	2433	3643	5479	2433	3643	5229	8087
1	614	893	1118	1321	2433	3643	5229	8087	3047	4536	6347	9408
3	977	0	1184	1098	3047	4536	6347	9408	4024	4536	7531	10506
2	812	0	1298	1534	4024	4836	7531	10506	4836	4836	8829	12040
MAKESPAN (Detik)									12040			

Dari hasil perhitungan *makespan* sebelumnya telah didapatkan bahwa total *makespan* metode usulan algoritma CEGA sebesar 10829 detik, sedangkan metode yang dipakai perusahaan didapatkan nilai *makespan* sebesar 12040 detik. Maka untuk menghitung besar parameter *efisiensi* adalah sebagai berikut:

$$Efisiensi = \frac{Z_{Makespan-Perusahaan} - Z_{Makespan-CEGA}}{Z_{Makespan-Persahaan}} \times 100\% \tag{9}$$

Berikut ini hasil perbandingan metode usulan dengan metode yang dipakai perusahaan yang ditampilkan pada Tabel 13.

Tabel 13. Perbandingan Metode Usulan Dengan Metode Yang Dipakai Perusahaan

NO	Parameter	Usulan (CEGA)	Perusahaan
1	Urutan Pekerjaan	2-1-3-4-5	4-5-1-3-2
2	Nilai makespan (Detik)	10829	12040
3	Selisih waktu (detik)	1211	
4	Efisiensi (%)	10.06%	

Berdasarkan hasil perhitungan nilai efisiensi, didapatkan nilai sebesar 10,06% maka algoritma CEGA menunjukkan lebih efisien dari segi *makespan* dibanding dengan metode yang dipakai perusahaan.

KESIMPULAN

Berdasarkan hasil penelitian ini bahwa didapatkan urutan pekerjaan yang optimal terletak pada kondisi urutan pekerjaan 2-1-3-4-5 dengan nilai *makespan* sebesar 10829 detik. Jika dibandingkan dengan nilai *makespan* antara metode CEGA dengan perusahaan tersebut akan didapatkan efisiensi sebesar 10,06%. Sehingga diharapkan hasil penelitian ini menjadi bahan pertimbangan perusahaan untuk mampu menanggapi masalah penjadwalan yang dihadapi oleh perusahaan.

DAFTAR PUSTAKA

- Baker, K., 1974, *Introduction to Sequencing and Scheduling*, John Wiley and Sons Inc., Singapore.
- Berlianti, I., dan Arifin, M., 2010, Teknik-teknik Optimasi *Heuristik*, Graha Ilmu. Yogyakarta.
- Budiman M.A., 2010, *Pendekatan Cross Entropy-Genetic Algorithm Untuk Permasalahan Penjadwalan Job Shop Tanpa Waktu Tunggu Pada Banyak Mesin*, Surabaya: Institut Teknologi Sepuluh Nopember.
- Hanka, M. K. R., dan Santosa, B., 2013. *Pengembangan Algoritma Hybrid Cross Entropy-Genetic Algorithm Pada Permasalahan Multiobjective Job Shop Scheduling Untuk Minimasi Makespan Dan Mean Flow Time*, Tugas Akhir : Institut Teknologi Sepuluh Nopember.
- Nurkhalida, L., dan Santosa, B., 2012, *Pendekatan Cross Entropy-Genetic Algorithm Pada Permasalahan Multi Objective Job Shop Scheduling*, UPT. Perpustakaan Institut Teknologi Sepuluh Nopember Surabaya.
- Santosa B., dan Willy, P., 2011, *Metoda Metaheuristik Konsep dan Implementasi*, Guna Widya, Surabaya.
- Widodo, D.S., Santoso, P. B., dan Siswanto, E., 2014, *Pendekatan Algoritma Cross Entropy-Genetic Algorithm Untuk Menurunkan Makespan Pada Penjadwalan Flow Shop*, Journal of Engineering and Management in Industrial System Vol. 2.