

# PENJADWALAN FLOW SHOP DENGAN PENERAPAN CROSS ENTROPY-GENETIC ALGORITHM (CEGA) UNTUK MEMINIMASI MAKESPAN

Hasan Bashori<sup>1</sup>, Pratikto<sup>2</sup>, Sugiono<sup>3</sup>

<sup>1,2,3</sup>Universitas Brawijaya, Fakultas Teknik Mesin, Malang 65145, Indonesia

**ABSTRACT** In this study done on leather shoe company that uses flow shop scheduling strategies. Where the purpose of this study is to minimize makespan is the total time needed to complete the entire job. This is done because of the large demand for leather shoes that exceeds the production capacity and production scheduling by the company based on intuition. In solving the production scheduling problem that occurs in the company, Cross Entropy algorithm used method-Genetic Algorithm (CEGA) to minimize the makespan. From the results of calculations with CEGA method sequences obtained by scheduling makespan value of 5822 seconds with an efficiency of 6.79% when compared with the method of the company with sebesar makespan value of 6246 seconds.

**Keywords:** Cross Entropy Algorithm-Genetic Algorithm (CEGA), Flow Shop Scheduling, Makespan Minimization

## 1. PENDAHULUAN

Menurut [1] penjadwalan dalam sistem produksi dapat diartikan sebagai kegiatan mengalokasikan sejumlah pekerjaan ke sejumlah sumber daya yang ada. Penjadwalan sebagai proses pengambilan keputusan memiliki peranan penting dalam kegiatan produksi dan informasi [2]. Oleh sebab itu, dibutuhkan pengembangan dan pendekatan untuk mendapatkan penjadwalan yang efektif dan efisien [3].

Penjadwalan produksi *flow shop* merupakan salah satu kegiatan perencanaan yang terdapat pada perusahaan manufaktur. Dimana penjadwalan produksi melibatkan  $n$  *job* (jenis pekerjaan) dan  $m$  mesin (jenis mesin) yang dalam proses produksinya, produk mendatangi mesin dengan urutan tahap yang sama dan pada setiap tahap terdiri atas 1 buah mesin yang mana setiap *job* yang dikerjakan mengandung informasi tentang jenis produk. Pada dasarnya penjadwalan produksi yang menggunakan strategi *flow shop* bertujuan untuk menyelesaikan serangkaian pekerjaan (*job*) berdasar pada urutan proses.

Perusahaan sepatu ini merupakan perusahaan yang bergerak pada bidang pembuatan sepatu kulit dan berdiri sejak Tahun 2009. Perusahaan ini terletak di daerah Jawa Timur. Dalam proses aliran produksi untuk menyelesaikan pekerjaan (*job*) menggunakan strategi *flow shop*.

Dari hasil wawancara yang telah dilakukan, diperoleh sebuah informasi tentang permasalahan yang terjadi pada perusahaan yaitu tidak adanya perencanaan penjadwalan produksi dan tingginya tingkat permintaan yang melebihi kapasitas produksi, sedangkan perusahaan diharapkan untuk dapat mengoptimalkan penjadwalan *job* dalam upaya pemenuhan permintaan *customer* dan meningkatkan kapasitas produksi.

Setelah mengetahui permasalahan yang terjadi pada perusahaan, maka diperlukan sebuah penjadwalan *job* dalam upaya menyelesaikan masalah untuk dapat meminimasi *makespan*. Yang mana pengertian dari *makespan* itu sendiri adalah total waktu yang dibutuhkan untuk menyelesaikan seluruh *job*. Sehingga dengan adanya proses penjadwalan produksi yang sistematis diharapkan bisa membantu perusahaan dapat meningkatkan jumlah produksi serta dapat menyelesaikan *job* dalam upaya memenuhi kebutuhan *customer*.

Pada penelitian ini, peneliti mencoba memberikan sebuah solusi pada permasalahan tersebut dengan menggunakan metode *metaheuristic* model algoritma untuk memperoleh penjadwalan yang sistematis dan efisien. Model algoritma yang digunakan dalam penelitian ini dengan cara melakukan penggabungan algoritma *Cross Entropy-Genetic Algorithm* (CEGA) yang bertujuan untuk memperoleh urutan penjadwalan *job* yang optimal dalam upaya meminimasi *makespan*.

Metode *Cross Entropy* ini nantinya akan digabungkan dengan *Genetic Algorithm* dengan menentukan inisialisasi parameter sebagai acuan

\* Corresponding author: Hasan Bashori, Pratikto, Sugiono  
[bashori86@gmail.com](mailto:bashori86@gmail.com)

Published online at <http://JEMIS.ub.ac.id/>

Copyright ©2015 JTI UB Publishing. All Rights Reserved

dalam proses perhitungan. Setelah itu dilakukan pembangkitan sampel algoritma CEGA yang diambil secara *random* untuk menentukan urutan *job*. Setelah diketahui urutan dari masing-masing *job* maka dilakukan penetapan fungsi tujuan, yaitu mencari nilai *makespan* yang paling minimum. Kemudian jika sudah ditemukan nilai *makespan* yang paling minimum maka akan dijadikan sebagai sampel elit, yang berarti sebagai induk untuk mencari anakan selanjutnya dengan cara proses mekanisme pindah silang dan mutasi. Tujuan konsep ini berfungsi sebagai verifikasi solusi optimal. Sehingga menghindari kemungkinan pencarian solusi terjebak pada sampel elit algoritma *Cross Entropy* ketika berada di area lokal optimal dengan menambah mekanisme pindah silang dan mutasi dari *Genetic Algorithm*. Dengan elitisme, sampel dengan nilai solusi terbaik akan disimpan agar dapat muncul pada sampel diiterasi berikutnya. Oleh sebab itu dengan adanya metode penjadwalan yang optimal diharapkan dapat membantu “CV. X” untuk menyelesaikan produksi tepat waktu sesuai dengan kebutuhan *customer*.

Adapun tujuan dari penelitian ini adalah merancang algoritma CEGA untuk memperoleh penjadwalan *job* yang optimal sehingga dapat meminimasi *makespan* dan membandingkan antara metode yang ada diperusahaan dengan metode algoritma CEGA.

## 2. METODE PENELITIAN

### Algoritma *Cross Entropy* (CE)

Secara garis besar, cara kerja algoritma *cross entropy* dapat digambarkan sebagai berikut [4] : strategi perolehan informasi oleh algoritma ini adalah bagaimana mengambil sampel yang tepat dari ruang lingkup permasalahan dan mendapatkan gambaran distribusi dari solusi yang bagus. Dengan distribusi ini akan di-*generate* kandidat solusi baru. 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.

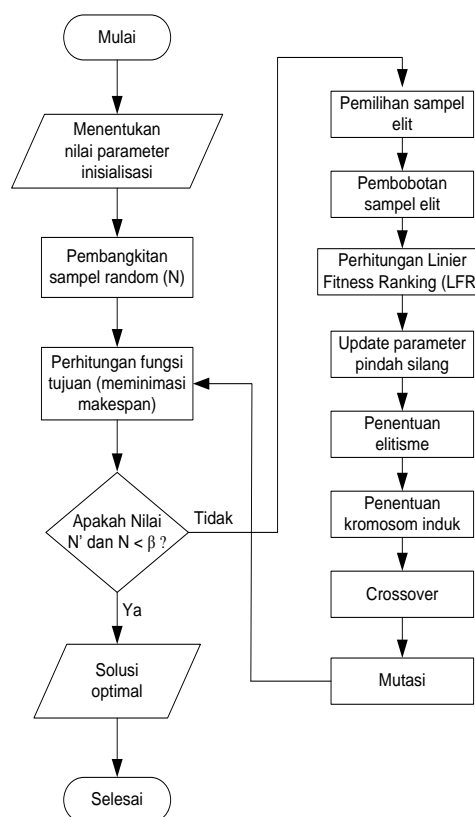
### Genetic Algorithm (GA)

Menurut [4] secara umum cara kerja dari *genetic algorithm* dapat dijelaskan sebagai berikut: langkah awal adalah mendapatkan populasi dengan individu yang dibangkitkan secara *random*. Hal ini akan berlaku pada setiap iterasi algoritma, yang dalam kasus *genetic algorithm* diistilahkan sebagai generasi populasi sampel. Pada setiap generasi,

*fitness* dari masing-masing individu acak tadi di evaluasi, kemudian dipilih individu dengan nilai *fitness* terbaik. Individu terpilih ini (kromosom induk) selanjutnya akan dimodifikasi untuk membentuk populasi baru. Modifikasi terdiri dari dua jenis mekanisme, yaitu *crossover* dan mutasi akan didapatkan kromosom anak dengan jumlah sama dengan induknya. Populasi baru yang terdiri dari kromosom anak dan induk kemudian digunakan dalam generasi berikutnya.

### Pengembangan Algoritma *Cross Entropy-Genetic Algorithm* (CEGA) Sebagai Konsep Solusi

Dengan melakukan penggabungan metode algoritma *Cross Entropy* dan *Genetic Algorithm* diharapkan dapat memperoleh sebuah konsep solusi yang lebih baik. Pada Gambar 1 berikut ini akan dijelaskan mengenai alur pengembangan metode algoritma *Cross Entropy-Genetic Algorithm* (CEGA) yang dikutip dari penelitian [5].



Gambar 1. Flowchart Algoritma *Cross Entropy-Genetic Algorithm* (CEGA)

## 3. HASIL DAN PEMBAHASAN

### Perhitungan Algoritma CEGA

#### Tahap 1: Inisialisasi Parameter

Tahap inisialisasi parameter algoritma CEGA yang akan digunakan pada contoh perhitungan ini adalah:

- Jumlah sampel yang dibangkitkan ( $N$ ) = 7

- b. Parameter kejarangan ( $\rho$ )= 0.03
- c. Koefisien penghalusan ( $\alpha$ )= 0.5
- d. Parameter pindah silang ( $P_{ps}$ ) = 1
- e. Parameter pemberhentian ( $\beta$ )= 0.0001

**Tahap 2: Membangkitkan Sampel**

Bentuk sampel yang mewakili prioritas urutan *job* dari seluruh operasi pengerjaan *job* ini dibangkitkan secara *random*. Sehingga dalam penelitian ini akan dilakukan pembangkitan bilangan *random* yang membentuk 7 sampel sebagai berikut:

- Y1= 1-2-3-4                      Y5= 2-4-1-3
- Y2= 4-3-2-1                    Y6= 1-3-4-2
- Y3= 3-2-1-4                    Y7= 2-1-3-4
- Y4= 2-3-4-1

Sampel yang di ambil secara *random* berupa urutan pengerjaan *job*, dimana sampel Y1 urutan pengerjaan *job* yaitu *job* 1, *job* 2, *job* 3, dan *job* 4, serta sampel Y2 urutan pengerjaan *job* yaitu *job* 4, *job* 3, *job* 2, dan *job* 1. Begitu juga untuk sampel berikutnya.

**Tahap 3: Menghitung Fungsi Tujuan**

Tahap perhitungan fungsi tujuan dihitung berdasarkan nilai *makespan*, berikut ini akan diberikan contoh perhitungan satu sampel dan untuk sampel yang lain dihitung dengan langkah yang sama juga. Sampel yang digunakan adalah sampel kelima (Y5) = 2 – 4 – 1 - 3. Urutan pertama dari sampel 2-4-1-3 adalah *job* ke 2.

Dalam menentukan nilai waktu mulai dan waktu selesai dalam proses perhitungan fungsi tujuan menggunakan syarat-syarat berikut ini:

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

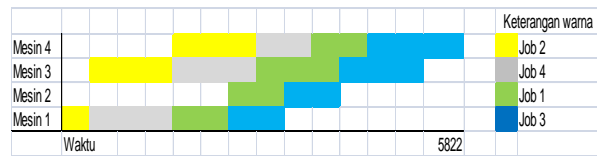
Yang mana dalam operasi prasyarat memiliki hubungan dalam satu *job*, sedangkan operasi pendahulu memiliki hubungan dalam operasi mesin. Berikut hasil perhitungan *makespan* pada penjadwalan *job* sampel Y5 yaitu 2 – 4 – 1 – 3, dan total *makespan* yang didapat sebesar 5822 detik. Untuk hasil perhitungan seperti pada Tabel 1.

**Tabel 1.** Nilai Fungsi Tujuan *Job* Sampel Y5

URUTAN	WAKTU PROSES (Detik)								MULA (Detik)								SELESAI (Detik)																														
	M1	M2	M3	M4	M5	M6	M7	M8	M1	M2	M3	M4	M5	M6	M7	M8	M1	M2	M3	M4	M5	M6	M7	M8																							
2	302	303	151	0	482	181	181	871	0	302	605	756	756	1238	1418	1610	302	605	756	756	1238	1418	1610	2471																							
4	331	342	168	0	541	183	183	1038	302	633	975	1143	1238	1779	1982	2471	633	975	1143	1143	1779	1982	2145	3479																							
1	381	602	181	121	541	181	182	1021	633	1024	1626	1807	1928	2469	2650	3479	1024	1626	1807	1808	2469	2650	2832	4500																							
3	541	721	183	123	682	182	181	1322	1024	1626	2347	2530	2653	3315	3487	4500	1626	2347	2530	2653	3315	3487	3678	5822																							
MAKESPAN (Detik)																								5822																							

Sedangkan rumus untuk menghitung *Z* (*Makespan*) adalah  $Z = \max_{1 \leq j \leq n} (F_j)$ , maka nilai

Maksimum dari *flow time* ( $F_j$ ) yaitu 5822 detik. Sehingga nilai *makespan* (*Z*) sebesar 5822 detik. Gambar 2berikut ini adalah model Gambar *Gantt Chart* sampel Y5 (2-4-1-3).



**Gambar 2.** Gambar *Gantt Chart* Sampel Y5

Hasil perhitungan nilai *makespan* dari ke tujuh sampel seperti pada Tabel 2.

**Tabel 2.** Rekapitulasi Hasil Perhitungan *Makespan* Untuk Semua Sampel

No	Sampel	Urutan Job	Makespan
1	Y1	1 2 3 4	6421
2	Y2	4 3 2 1	6138
3	Y3	3 2 1 4	6815
4	Y4	2 3 4 1	6246
5	Y5	2 4 1 3	5822
6	Y6	1 3 4 2	6421
7	Y7	2 1 3 4	5852

**Tahap 4: Menentukan Sampel Elit**

Dalam penentuan sampel elit, nilai *makespan* yang diperoleh dari semua sampel di urutkan mulai dari yang terkecil sampai terbesar. Pada tahap inisialisasi sebelumnya telah dipilih nilai *rho* sebesar 0.03, maka rumus jumlah sampel elit adalah  $N \times \rho = 7 \times 0.03 = 0.21 \approx 1$  [4]. Dari tabel 3 berikut ini akan diambil satu sampel teratas dengan nilai *makespan* terkecil sebagai sampel elit, yaitu sampel ke-5 (Y5) dengan urutan prioritas *job* yaitu 2-4-1-3.

**Tabel 3.** Hasil Rekapitulasi Pengurutan *Makespan*

No	Sampel	Urutan Job	Makespan
Y1	Y5	2 4 1 3	5822
Y2	Y7	2 1 3 4	5852
Y3	Y2	4 3 2 1	6138
Y4	Y4	2 3 4 1	6242
Y5	Y1	1 2 3 4	6421
Y6	Y6	1 3 4 2	6421
Y7	Y3	3 2 1 4	6815

**Tahap 5: Pembobotan Sampel Elit**

Pembobotan sampel elit diperoleh dari evaluasi terhadap nilai terbaik pada iterasi sebelumnya [4]. Karena dalam proses perhitungan menentukan sampel elit hanya terdapat 1 sampel elit saja, maka bobot sampel elit = 1.

**Tahap 6: Menghitung Linier Fitness Rangking (LFR)**

Nilai LFR digunakan untuk pemilihan induk pada proses pindah silang (*crossover*). LFR diperoleh melalui rumus [4]:

$$LFR(I(N-i+1)) = F_{max} - (F_{max} - F_{min}) * ((i-1)/(N-1)) \quad (\text{Pers. 1})$$

dengan  $F_{max} = 1/Z(1) = 0,0001718$  dan  $F_{min} = 1/Z(N) = 1/Z(7) = 0,0001467$ , dari hasil nilai *makespan* yang sudah di urutkan, maka LFR dari ke tujuh sampel adalah:

$$Y1 = Y5 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((1-1)/(7-1))) = 0,0001718$$

$$Y2 = Y7 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((2-1)/(7-1))) = 0,0001676$$

$$Y3 = Y2 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((3-1)/(7-1))) = 0,0001634$$

$$Y4 = Y4 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((4-1)/(7-1))) = 0,0001592$$

$$Y5 = Y1 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((5-1)/(7-1))) = 0,0001551$$

$$Y6 = Y6 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((6-1)/(7-1))) = 0,0001509$$

$$Y7 = Y3 \rightarrow LFR = 0,0001718 - ((0,0001718 - 0,0001467) * ((7-1)/(7-1))) = 0,0001467$$

**Tahap 7: Update Parameter Pindah Silang**

*Update* parameter dilakukan agar memperoleh nilai parameter yang *update* untuk evaluasi kriteria pemberhentian. Pada tiap iterasi dilakukan *update* parameter pindah silang. Semakin besar nilai parameter pindah silang maka semakin banyak jumlah sampel yang akan mengalami pindah silang. Untuk rumus *update* parameter pindah silang sebagai berikut [4]:

$$P_{ps(i)} = (1-\alpha) * u + (P_{ps(i+1)}) * \alpha \quad (\text{Pers. 2})$$

dengan  $u$  adalah:

$$u = \frac{\bar{Z}_e}{2 * Z_{best}} \quad (\text{Pers. 3})$$

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

$$u = \frac{\bar{Z}_e}{2 * Z_{best}} = \frac{5822}{2 * 5822} = 0,5$$

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

**Tahap 8: Elitisme**

Tahap elitisme bertujuan untuk menyimpan sampel dengan nilai fungsi tujuan terbaik pada tiap iterasi, yang nantinya sampel akan muncul kembali pada sampel di iterasi berikutnya. Maka dari itu untuk menjaga agar individu bernilai fitness tertinggi tidak hilang selama evolusi [4]. Jumlah sampel yang di elitisme sebanyak satu, karena dari hasil dari perhitungan sampel elit dihasilkan nilai 1. Pada kasus ini, sampel yang di elitisme adalah nomer Y1=2-4-1-3.

**Tahap 9: Proses Pemilihan Induk Pindah Silang**

Proses pemilihan dua buah kromosom sebagai induk yang akan dipindah silang dilakukan secara proporsional sesuai dengan dengan nilai *fitness*-nya. Dengan menggunakan mekanisme *roulette wheel*, akan dilakukan pemilihan induk 1 dari sampel elit dan induk 2 dari sampel keseluruhan.

**1) Pemilihan induk 1**

Pemilihan induk 1 di ambil dari nilai sampel elit yang pada proses perhitungan sebelumnya memiliki nilai fungsi tujuan terbaik dari mekanisme elitisme diperoleh pada sampel nomer Y1=2-4-1-3.

**2) Pemilihan induk 2**

Pemilihan induk 2 dipilih berdasar pada nilai evaluasi dari sampel keseluruhan dengan memakai nilai LFR dari sampel yang dievaluasi. Apabila nilai perbandingan antara kumulatif LFR dan total LFR lebih besar dari nilai *random* yang dibangkitkan, maka sampel tersebut menjadi induk 2.

$$\begin{aligned} \text{Total LFR} &= 0,0001718 + 0,0001676 + 0,0001634 \\ &+ 0,0001592 + 0,0001551 + 0,0001509 \\ &+ 0,0001467 = 0,0011147 \end{aligned}$$

Hasil Kumulatif LFR total dibandingkan nilai *random* yang dibangkitkan (0.4324) berdasarkan nomer urutan nilai *makespan* minimum adalah:

$$Y1 = 0,0001718 / 0,0011147 = 0,154 < 0,4324$$

$$Y2 = (0,0001718 + 0,0001676) / 0,0011147 = 0,304 < 0,4324$$

$$Y3 = (0,0001718 + 0,0001676 + 0,0001634) / 0,0011147 = 0,451 > 0,4324$$

$$Y4 = (0,0001718 + 0,0001676 + 0,0001634 + 0,0001592) / 0,0011147 = 0,594 > 0,4324$$

$$Y5 = (0,0001718 + 0,0001676 + 0,0001634 + 0,0001592 + 0,0001551) / 0,0011147 = 0,733 > 0,4324$$

$$Y6 = (0.0001718 + 0.0001676 + 0.0001634 + 0.0001592 + 0.0001551 + 0.0001509) / 0.0011147 = 0.868 > 0.4324$$

$$Y7 = (0.0001718 + 0.0001676 + 0.0001634 + 0.0001592 + 0.0001551 + 0.0001509 + 0.0001467) / 0.0011147 = 1 > 0.4324$$

Berdasarkan pada hasil perhitungan diatas, diketahui bahwa sampel nomer Y1 dijadikan induk 1 sedangkan sampel Y3, Y4, Y5, Y6, dan Y7 ditetapkan sebagai induk 2.

**Tahap 10: Crossover (Pindah Silang)**

Crossover merupakan bagian tahapan dari algoritma genetik yang menyilangkan 2 induk untuk membentuk kromosom baru untuk menghasilkan individu baru yang lebih baik. Metode yang digunakan adalah 2-point order crossover. Dari hasil langkah pemilihan induk pindah silang diperoleh bahwa Y1 ditetapkan sebagai induk 1 dan Y3, Y4, Y5, Y6, dan Y7, sebagai induk 2.

Apabila pada pembangkitan bilangan random diperoleh bilangan random (R) lebih kecil dari parameter pindah silang (P<sub>ps</sub>) atau R < P<sub>ps</sub>, maka akan dilakukan pindah silang terhadap kedua kromosom induk tersebut. Namun jika pembangkitan bilangan random (R) lebih besar dari parameter pindah silang (P<sub>ps</sub>) atau R > P<sub>ps</sub>, maka tidak terjadi pindah silang terhadap kedua kromosom induk. Maka dari itu, untuk mengetahui sampel yang mengalami cross over dapat dilihat pada Tabel 4.

**Tabel 4. Menentukan Crossover**

No	Sampel	Random (R)	P <sub>ps</sub>	Kesimpulan
1	Y5= 2-4-1-3	0.8274	0.75	Tidak dilakukan Cross Over
2	Y7= 2-1-3-4	0.5972	0.75	Dilakukan Cross Over
3	Y2= 4-3-2-1	0.5807	0.75	Dilakukan Cross Over
4	Y4= 2-3-4-1	0.2844	0.75	Dilakukan Cross Over
5	Y1= 1-2-3-4	0.3082	0.75	Dilakukan Cross Over
6	Y6= 1-3-4-2	0.7109	0.75	Dilakukan Cross Over
7	Y3= 3-2-1-4	0.5799	0.75	Dilakukan Cross Over

Pada sampel nomer Y2 dan Y3 diperoleh hasil bilangan random yang telah dibangkitkan dengan nilai lebih kecil dari P<sub>ps</sub> (0.75) maka akan dilakukan pindah silang antara induk nomer Y1 dengan Y3. Dalam penentuan bagian dari sampel induk yang akan dilakukan penukaran, maka akan dilakukan mekanisme berikut: dengan melakukan pembangkitan dua bilangan random, diperoleh bilangan 0.2354 dan 0.6968 (Tabel 5). Kemudian nilai random tersebut dikonversi menjadi nilai bulat, dan dipakai sebagai pembatas bagian sampel yang akan di pindah antar induk.

$$r_i = \text{ceil}(\text{random} * n) \dots\dots(4)$$

$$r1 = \text{ceil}(0.2345 * 4) = 1 \text{ maka } p1 = 1$$

$$r2 = \text{ceil}(0.6968 * 4) = 3 \text{ maka } p2 = 3$$

$$\text{Induk 1} = 2 \mid 4 \mid 1 \mid 3$$

$$\text{Induk 2} = 4 \mid 3 \mid 2 \mid 1$$

Setelah itu angka didalam kurung ditukar antar induk dan menjadi seperti berikut ini:

$$\text{Calon anak 1} = \dots 3 \mid 2 \dots$$

$$\text{Calon anak 2} = \dots 4 \mid 1 \dots$$

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

$$\text{Anak 1} = 4 - 3 - 2 - 1 \rightarrow \text{menggantikan Y2 lama}$$

$$\text{Anak 2} = 2 - 4 - 1 - 3 \rightarrow \text{menggantikan Y3 lama}$$

Selanjutnya ulangi dengan mengevaluasi urutan sampel ke 4, 5, 6, dan 7 dengan menggunakan perhitungan yang sama dengan awal, diketahui bahwa induk1 = Y1 dan induk2 = Y4, Y5, Y6, dan Y7. Dengan menggunakan random lebih kecil dari P<sub>ps</sub> (0.75), maka terjadi pindah silang. Induk1 dan induk2 digunakan untuk mengganti urutan ke- 4, ke- 5, ke-6 dan ke-7 (hasil selengkapnya dapat dilihat pada Tabel 5 dan Tabel 6) , maka populasi baru menjadi sebagai berikut:

$$Y1 = 2 - 4 - 1 - 3 \qquad Y5 = 2 - 4 - 3 - 1$$

$$Y2 = 4 - 3 - 2 - 1 \qquad Y6 = 1 - 3 - 4 - 2$$

$$Y3 = 2 - 4 - 1 - 3 \qquad Y7 = 4 - 2 - 1 - 3$$

$$Y4 = 2 - 3 - 1 - 4$$

**Tabel 5. Proses Crossover**

Induk Cross Over	Urutan Job	Random	r <sub>i</sub> = ceil (random * n) dan n=4	Cell Yg dicross over	Hasil Cross over	Keterangan	
1	2	3	5	6	7	8	
Y1 dg Y3	2-4-1-3	r1	0.2354	0.9416 = 1	2   4 - 1   3	4-3-2-1	Sampel Y2
	4-3-2-1	r2	0.6968	2.7872 = 3	4   3 - 2   1	2-4-1-3	Sampel Y3
Y1 dg Y4	2-4-1-3	r1	0.2542	1.0168 = 1	2   4   1-3	2-3-1-4	Sampel Y4
	2-3-4-1	r2	0.4321	1.7284 = 2	2   3   4-1	2-4-3-1	-
Y1 dg Y5	2-4-1-3	r1	0.5032	2.0128 = 2	2-4   1   3	2-4-3-1	Sampel Y5
	1-2-3-4	r2	0.6821	2.7284 = 3	1-2   3   4	3-2-1-4	-
Y1 dg Y6	2-4-1-3	r1	0.2281	0.9124 = 1	2   4 - 1 - 3	1-3-4-2	Sampel Y6
	1-3-4-2	r2	0.8842	3.5368 = 4	1   3 - 4 - 2	2-4-1-3	-
Y1 dg Y7	2-4-1-3	r1	0.2384	0.9536 = 1	2   4   1-3	4-2-1-3	Sampel Y7
	1-3-4-2	r2	0.4584	1.8336 = 2	1   3 - 4 - 2	3-4-1-2	-

**Tabel 6. Hasil Crossover**

No	Sampel	Random (R)	P <sub>ps</sub>	Kesimpulan	Hasil Cross over	Keterangan
1	Y5= 2-4-1-3	0.8274	0.75	Tidak dilakukan Cross Over	2-4-1-3	Diambil dari Kromosom Induk Y1
2	Y7= 2-1-3-4	0.5972	0.75	Dilakukan Cross Over	4-3-2-1	Hasil cross over induk Y1 dan Y3
3	Y2= 4-3-2-1	0.5807	0.75	Dilakukan Cross Over	2-4-1-3	Hasil cross over induk Y1 dan Y3
4	Y4= 2-3-4-1	0.2844	0.75	Dilakukan Cross Over	2-3-1-4	Hasil cross over induk Y1 dan Y4
5	Y1= 1-2-3-4	0.3082	0.75	Dilakukan Cross Over	2-4-3-1	Hasil cross over induk Y1 dan Y5
6	Y6= 1-3-4-2	0.7109	0.75	Dilakukan Cross Over	1-3-4-2	Hasil cross over induk Y1 dan Y6
7	Y3= 3-2-1-4	0.5799	0.75	Dilakukan Cross Over	4-2-1-3	Hasil cross over induk Y1 dan Y7

**Tahap 11: Mutasi**

Proses mutasi dilakukan untuk memunculkan individu baru yang tidak sama dengan individu yang sudah ada. Probabilitas mutasi (P<sub>m</sub>) disini akan menentukan kromosom mana yang akan mengalami perubahan gen,

semakin besar nilai *probabilitas* mutasi maka semakin banyak kromosom dalam populasi yang akan mengalami mutasi [6]. Proses mutasi dipilih secara *random* dan gen pada site tersebut akan diubah nilainya. Angka *random* akan dibangkitkan dengan batasan 0 sampai 1. Jika angka *random* (R) tersebut lebih kecil dari parameter mutasi ( $P_m$ ) maka digit gen akan diganti, dan jika angka *random* (R) tersebut lebih besar dari parameter mutasi ( $P_m$ ) maka digit gen tidak akan diganti.

Nilai parameter mutasi ditentukan dengan rumus sebagai berikut [7]:

$$P_m = \frac{P - ps}{2} = \frac{0.75}{2} = 0.375 \quad \dots\dots(5)$$

Setelah diperoleh nilai parameter mutasi sebesar 0.375 dan N sebesar 7. Maka akan diperoleh jumlah sampel kromosom yang akan mengalami mutasi yaitu:

$$N_a = P_m \times N = 0.375 \times 7 = 2.625 \approx 3; \text{ sehingga terdapat 3 sampel kromosom yang akan mengalami mutasi.}$$

Agar dapat mengetahui sampel kromosom mana yang akan mengalami mutasi maka dibangkitkan bilangan *random*. Tabel 7 adalah hasil pembangkitan bilangan *random* serta kromosom yang akan mengalami proses mutasi.

**Tabel 7. Penentuan Mutasi Pada Kromosom**

No	Hasil Cros over	Random mutasi	Probabilitas Mutasi (Pm)	Keterangan
1	2	3	4	5
1	Y1 tetap			
2	4 - 3 - 2 - 1	0.3127	0.375	R < Pm maka dilakukan mutasi
3	2 - 4 - 1 - 3	0.3646	0.375	R < Pm maka dilakukan mutasi
4	2 - 3 - 1 - 4	0.0854	0.375	R < Pm maka dilakukan mutasi
5	2 - 4 - 3 - 1	0.2257	0.375	R < Pm maka dilakukan mutasi
6	1 - 3 - 4 - 2	0.6347	0.375	R > Pm maka tidak dilakukan mutasi
7	4 - 2 - 1 - 3	0.7706	0.375	R > Pm maka tidak dilakukan mutasi

Dari tabel diatas didapatkan kromosom Y2, Y3, Y4, dan Y5 yang mengalami mutasi, maka untuk tahap selanjutnya yakni melakukan mutasi pada gen ke 2 sampel kromosom tersebut. Berikut ini perhitungan mutasi yang akan diberikan pada contoh sampel kromosom Y2:

$rand1 = 0.1792; rand2 = 0.8863; n = 4$   
 $I, J = \text{ceil}(n * r_{ij}) \quad \text{(Pers. 6)}$   
 $I = \text{ceil}(4 * 0.1792) = 0.717 \approx 1$   
 $J = \text{ceil}(4 * 0.8863) = 3.454 \approx 4$   
 Keterangan jika nilai:  
 K = 1 : maka dilakukan *flip mutation* (membalik)  
 K = 2 : maka dilakukan *swap mutation* (menukar)  
 K = 3 : maka dilakukan *slide mutation* (menggесer)

$rand K = 0.5992$

$K = \text{ceil}(rand K * 3) \quad \text{(Pers. 7)}$   
 $K = \text{ceil}(0.5992 * 3) = 1,7976 \approx 2$  berarti dilakukan *swap mutation*  
 $Y2 = 4 - 1 - 2 - 3$  dilakukan *swap mutation*

Untuk mengetahui proses mutasi yang terjadi pada Y3, Y4, dan Y5, juga dilakukan langkah sama seperti perhitungan Y2. Maka diperoleh proses gen yang dimutasi pada sampel Y3, Y4, dan Y5 sebagai berikut:

$Y3 = 2 - 1 - 4 - 3$  dilakukan *flip mutation*  
 $Y4 = 2 - 4 - 3 - 1$  dilakukan *slide mutation*  
 $Y5 = 2 - 3 - 4 - 1$  dilakukan *swap mutation*

Setelah melakukan mutasi pada sampel Y2, Y3, Y4, dan Y5, maka terjadi perubahan berikut ini, dan lebih jelasnya bisa dilihat pada Tabel 8.

$Y2 = 4 - 1 - 2 - 3 \quad Y4 = 2 - 4 - 3 - 1$   
 $Y3 = 2 - 1 - 4 - 3 \quad Y5 = 2 - 3 - 4 - 1$

**Tabel 8. Tabel Hasil Mutasi**

No	Hasil Cros over	Random mutasi	Probabilitas Mutasi (Pm)	Keterangan	Random (r1/2)	Nilai I,J dengan n=4	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		
1	Y1 tetap									Y1=2 - 4 - 1 - 3		
2	4 - 3 - 2 - 1	0.3127	0.375	R < Pm maka dilakukan mutasi	r1	0.1792	I	0.717=1	0.5992	1/976=2	Swap mutation	Y2=4 - 1 - 2 - 3
					r2	0.8863	J	3.545=4				
3	2 - 4 - 1 - 3	0.3646	0.375	R < Pm maka dilakukan mutasi	r1	0.2114	I	0.846=1	0.3244	0.9732=1	Flip mutation	Y3=2 - 1 - 4 - 3
					r2	0.6867	J	2.747=3				
4	2 - 3 - 1 - 4	0.0854	0.375	R < Pm maka dilakukan mutasi	r1	0.1998	I	0.789=1	0.8876	2.6628=3	Slide mutation	Y4=2 - 4 - 3 - 1
					r2	0.9486	J	3.794=4				
5	2 - 4 - 3 - 1	0.2257	0.375	R < Pm maka dilakukan mutasi	r1	0.2585	I	1.026=1	0.5455	1.6365=2	Swap mutation	Y5=2 - 3 - 4 - 1
					r2	0.6433	J	2.573=3				
6	1 - 3 - 4 - 2	0.6347	0.375	R > Pm maka tidak dilakukan mutasi								Y6=1 - 3 - 4 - 2
7	4 - 2 - 1 - 3	0.7706	0.375	R > Pm maka tidak dilakukan mutasi								Y7=4 - 2 - 1 - 3

**Tahap 12: Perhitungan Nilai Fungsi Tujuan Dari Populasi Baru**

Dari proses hasil mutasi diperoleh sampel dengan anggota baru yang terdiri atas sampel hasil proses elitisme, pindah silang (*cros over*), dan mutasi. Berikut ini adalah rekap dari hasil perhitungan nilai fungsi tujuan untuk populasi baru pada Tabel 9.

**Tabel 9. Rekap Hasil Perhitungan Makespan Populasi Baru**

No	Urutan Job Hasil Mutasi	Makespan (Detik)
Y1	2 - 4 - 1 - 3	5822
Y2	4 - 1 - 2 - 3	5970
Y3	2 - 1 - 4 - 3	5852
Y4	2 - 4 - 3 - 1	5822
Y5	2 - 3 - 4 - 1	6246
Y6	1 - 3 - 4 - 2	6421
Y7	4 - 2 - 1 - 3	5970

Setelah melakukan perhitungan algoritma CEGA secara manual, maka dilakukan pengujian dengan menggunakan bantuan *software*. Adapun untuk hasil pengujian dengan menggunakan bantuan *software* seperti pada Gambar 3.



Gambar 3. Hasil Output Coding CEGA Solusi Optimal 1



Gambar 4. Hasil Output Coding CEGA Solusi Optimal 2

**Metode Yang Digunakan Perusahaan**

Dari permasalahan yang dihadapi oleh perusahaan, yang mana dalam penjadwalan produksi dalam perusahaan tidak memiliki metode penjadwalan. Untuk kondisi sekarang, metode yang digunakan perusahaan dengan cara intuisi atau perkiraan, sehingga dalam menyelesaikan urutan *job* pada kegiatan penjadwalan produksi dapat berubah sewaktu-waktu. Maka dari itu untuk memilih metode penjadwalan perusahaan dalam model pengerjaan *job* yang paling sering dikerjakan oleh perusahaan. Dimana model pengerjaan urutan *job* yang paling sering dilakukan oleh perusahaan yaitu *Job* 2-3-4-1. Sehingga diperoleh hasil perhitungan *makespan* sebesar 6246 detik. Untuk dapat mengetahui hasil perhitungan *makespan* seperti pada Tabel 10.

Tabel 10. Nilai *Makespan* Pada Perusahaan

URUTAN JOB	WAKTU PROSES (Detik)								MULAI (Detik)								SELESA (Detik)							
	M1	M2	M3	M4	M5	M7	M8	M1	M2	M3	M4	M5	M6	M7	M8	M1	M2	M3	M4	M5	M6	M7	M8	
2	302	303	151	0	492	181	871	0	302	605	756	756	1238	1419	1600	302	605	756	756	1238	1419	1600	2471	
3	541	721	183	123	682	182	181	1322	302	843	1594	1747	1870	2532	2714	2865	843	1594	1747	1870	2532	2714	2865	4217
4	331	342	168	0	541	183	183	1008	843	1594	1906	2074	2532	3073	3256	4217	1174	1906	2074	2074	3073	3256	3438	5225
1	381	602	181	121	541	181	182	1021	1174	1906	2508	2688	3073	3614	3795	5225	1565	2508	2688	2810	3614	3795	3677	6246
MAKESPAN (Detik)																6246								

Untuk dapat mengetahui *performance* parameter yang dipakai untuk dapat menentukan metode yang lebih baik, menggunakan pendekatan dengan cara mengukur nilai efisiensi. Efisiensi dipakai untuk bisa mengetahui seberapa besar perbedaan *makespan* yang dihasilkan oleh kedua metode. Rumus yang digunakan untuk mengetahui efisiensi adalah sebagai berikut:

$$Efisiensi = \frac{Z_{Makespan-Perusahaan} - Z_{Makespan-CEGA}}{Z_{Makespan-Perusahaan}} \times 100\% \quad \text{(Pers. 8)}$$

Pada perhitungan sebelumnya telah didapatkan nilai *makespan* dengan metode CEGA sebesar 5822 detik, sedangkan pada perusahaan nilai *makespan* yang didapat sebesar 6246 detik. Maka untuk mengetahui besar parameter *performance* efisiensi adalah:

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

$$Efisiensi = \frac{6246 \text{ det ik} - 5822 \text{ det ik}}{6246 \text{ det ik}} \times 100\%$$

$$Efisiensi = 6.79\%$$

Berdasarkan dari hasil perhitungan nilai efisiensi, didapatkan efisiensi sebesar 6.79%, maka diperoleh kesimpulan bahwa peran metode algoritma CEGA dalam penjadwalan pembuatan produk sepatu kulit lebih *efisien* dari sisi *makespan* dibandingkan dengan metode perusahaan.

**4. KESIMPULAN**

Berdasarkan dari hasil pengolahan dan analisis data, maka diperoleh nilai *makespan* sebesar 5822 detik pada metode CEGA dengan urutan *job* 2-4-1-3 dan *job* 2-4-3-1 sedangkan dari metode perusahaan didapatkan nilai *makespan* sebesar 6246 detik. Maka diperoleh selisih nilai *makespan* sebesar 424 detik atau berkisar 7 menit. Sedangkan dari hasil perhitungan nilai efisiensi, diperoleh nilai sebesar 6.79% yang berarti bahwa peran algoritma CEGA dalam meminimasi *makespan* pada penjadwalan pembuatan sepatu kulit lebih efisien dan lebih baik dibanding dengan metode di perusahaan saat ini. Sehingga terdapat 2 alternatif penjadwalan *job* yang dapat dilakukan oleh perusahaan.

**DAFTAR PUSTAKA**

[1] Bedworth, D.D. and J. E. Bailey. 1987. *Integrated Production Control System*. Toronto: John Wiley & Sons.  
 [2] Pinedo, M.L. 2011. *Scheduling Theory, Algorithm, and System*. 4<sup>th</sup> edition. New York: Springer.

- [3] Quan-Ke Pan, Ling Wang, dan Bao-Hua Zhao. 2008. *An improved Iterated Greedy Algorithm For The No-Wait Flow Shop Scheduling Problem With Makespan Criterion*. *International Journal Advanced Manufacturing Technology*. Vol. 38: hal. 778-786.
- [4] 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.
- [5] Widodo, D. S. 2014. Pendekatan Algoritma *Cross Entropy-Genetic Algorithm* Untuk Menurunkan *Makespan* Pada Penjadwalan *Flow Shop*. JEMIS.Vol. 2. No. 1.
- [6] Santosa B., dan Willy, P. 2011. Metoda Metaheuristik Konsep dan Implementasi. Guna Widya. Surabaya.
- [7] Hanka, M. K. R., dan Santosa, B. 2013. Pengembangan Algoritma *HybridCross Entropy-Genetic Algorithm* Pada Permasalahan *Multiobjective Job Shop Scheduling* Untuk Minimasi *Makespan* Dan *Mean Flow Time*. UPT. Perpustakaan Institut Teknologi Sepuluh Nopember.