

PENERAPAN ALGORITMA *CROSS ENTROPYGENETIC ALGORITHM* UNTUK OPTIMASI MAKESPAN PADA PENJADWALAN *FLOW SHOP*

PARANDITUS, EVI NOVIANI*, YUDHI

*Program Studi S1 Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura, Pontianak
email : evi.noviani@math.untan.ac.id*

Diterima 24 Oktober 2022 Direvisi 18 Januari 2023 Dipublikasikan 31 Januari 2023

Abstrak. Penjadwalan *flow shop* merupakan jenis penjadwalan produksi dimana setiap pekerjaan yang diproses seluruhnya mengalir pada proses yang sama, dengan tujuan untuk mencari urutan pekerjaan yang dapat mengoptimalkan nilai *makespan*. Penelitian ini bertujuan untuk mencari kombinasi urutan pekerjaan dengan nilai *makespan* yang optimal pada CV Bestone Indonesia. Permasalahan penjadwalan *flow shop* dapat diselesaikan dengan menggunakan *Cross EntropyGenetic Algorithm* (CEGA). Algoritma CEGA bekerja dengan cara membangkitkan sampel awal (N) secara acak, kemudian sampel selanjutnya diperoleh dari hasil proses *crossover* dan mutasi dengan parameter berupa parameter kejarangan (ρ), koefisien penghalusan (α), parameter *crossover* (P_c) dan parameter pemberhentian (β). Berdasarkan pengolahan data yang telah dilakukan menggunakan Algoritma CEGA, diperoleh urutan pekerjaan 3, pekerjaan 1, pekerjaan 4, pekerjaan 2 dan urutan pekerjaan 3, pekerjaan 4, pekerjaan 1, pekerjaan 2 serta nilai *makespan* sebesar 389116,52 detik.

Kata Kunci: Penjadwalan *flow shop*, *Makespan*, *Cross EntropyGenetic Algorithm*

1. Pendahuluan

Permasalahan dalam penyusunan jadwal produksi adalah sulitnya menentukan jadwal pekerjaan sehingga pekerjaan tersebut dapat selesai sesuai dengan batas waktu penyelesaian. Penjadwalan merupakan proses pengurutan pengerjaan setiap produk yang dikerjakan pada beberapa buah mesin. Proses pengurutan pengerjaan (*sequencing*) setiap produk melibatkan pengerjaan sejumlah elemen-elemen dasar yang disebut dengan operasi. Tiap operasi memerlukan alokasi sumber daya tertentu selama periode waktu tertentu yang sering disebut waktu proses, yaitu waktu pengerjaan yang diperlukan untuk melakukan operasi [1].

Berdasarkan pola alirannya, terdapat 2 jenis penjadwalan produksi yaitu penjadwalan *flow shop* dan penjadwalan *job shop*. Pada penjadwalan *flow shop* setiap

*penulis korespondensi

pekerjaan diproses di setiap mesin dengan urutan proses yang sama. Sedangkan pada penjadwalan *job shop*, setiap pekerjaan memiliki urutan proses yang berbeda [2]. Dalam penjadwalan *flow shop* terdapat beberapa pekerjaan (n) yang dijadwalkan pada beberapa mesin (m) dan waktu proses per unit pekerjaan ke- j pada mesin ke- i , dinotasikan sebagai t_{ij} (dengan $i = 1, \dots, m; j = 1, \dots, n$). Setiap pekerjaan memiliki urutan pemrosesan yang sama di setiap mesin, tujuan penjadwalannya adalah mencari urutan pemrosesan pekerjaan yang dapat meminimumkan total waktu yang diperlukan untuk menyelesaikan seluruh pekerjaan yang kemudian disebut *Makespan* [3].

CV Bestone Indonesia merupakan perusahaan yang bergerak di bidang hiasan dinding dengan bahan baku batu alam. Terletak di Kota Magelang, Provinsi Jawa Tengah. Jenis produk yang diproduksi adalah produk BEST 20 SH, produk BEST 47 A, produk BEST 47 dan produk BEST 50 SH. Dalam melakukan produksi, CV Bestone Indonesia menggunakan penjadwalan *flow shop* [4]. Akan tetapi, perusahaan ini menerapkan metode intuisi atau perkiraan untuk menentukan produk mana yang akan dikerjakan terlebih dahulu. Metode intuisi tidak mempertimbangkan nilai *makespan*, keterlambatan penyelesaian pekerjaan, dan batas waktu pesanan. Sehingga, perusahaan kesulitan dalam membuat jadwal produksi ketika banyaknya permintaan dengan jenis produk yang beragam. Oleh karena itu, diperlukan metode yang efektif untuk dapat mengoptimalkan nilai *makespan* sehingga proses produksi dapat meningkat dan permintaan produk dapat terpenuhi [4].

Masalah penjadwalan produksi seperti pada CV Bestone Indonesia, dapat diselesaikan dengan menggunakan beberapa metode diantaranya adalah Algoritma *Cross Entropy* (CE) dan *Genetic Algorithm* (GA). Algoritma CE pertama kali diperkenalkan pada tahun 1997 oleh Rubinstein yang bertujuan untuk mendapatkan urutan pekerjaan yang optimal dengan cepat melalui proses iterasi [5,11]. Sedangkan, GA merupakan metode pencarian yang memanfaatkan mekanisme operasi genetika dan seleksi alam [7,12,13,15]. penggabungan Algoritma CE dengan GA yang kemudian disebut Algoritma *Cross Entropy-Genetic Algorithm* (CEGA) telah diterapkan pada penjadwalan *flow shop* dengan menentukan nilai *makespan* yang terkecil [5,9,10]. Algoritma CEGA juga dapat diterapkan pada permasalahan penjadwalan pekerjaan dengan beberapa fungsi tujuan, seperti mengoptimalkan nilai *makespan* dan total *weighted tardiness* (TWT)[6,14].

Pada penelitian ini, penjadwalan *flow shop* pada CV Bestone Indonesia menggunakan Algoritma CEGA untuk menentukan urutan pekerjaan dan nilai *makespan* yang optimal. Proses penjadwalan *flow shop* dimulai dengan inisialisasi parameter dan membangkitkan sampel awal dengan bilangan acak. Setiap sampel kemudian dievaluasi dengan menghitung nilai *makespan*. Sampel selanjutnya diperoleh dari proses *crossover* dan mutasi. Setelah diperoleh sampel baru dari proses mutasi, kemudian dilakukan perhitungan fungsi tujuan (*makespan*) sampel baru hingga diperoleh urutan pekerjaan dengan nilai *makespan* yang paling minimum.

2. Algoritma *Cross Entropy Genetic Algorithm* (CEGA)

Algoritma CE bekerja dengan cara mengambil sampel awal secara acak dari ruang lingkup permasalahan. Sampel yang diperoleh kemudian dievaluasi dengan menghitung nilai fungsi tujuan. Selanjutnya, parameter diperbaharui berdasarkan sampel terbaik, dengan tujuan untuk memperoleh solusi yang lebih baik pada iterasi selanjutnya. Proses ini dilakukan secara berulang hingga diperoleh sampel yang optimal [6]. Sedangkan, Genetic Algorithm (GA) bekerja dengan membangkitkan sekumpulan kromosom secara acak untuk generasi awal. Selanjutnya, setiap kromosom akan dievaluasi berdasarkan nilai *fitness* dari masing-masing kromosom yang telah diperoleh, kemudian dipilih kromosom dengan nilai *fitness* terbaik. Kromosom yang terpilih (kromosom induk) selanjutnya ditukar dengan kromosom yang lain untuk membentuk kromosom baru melalui operasi genetika (*Crossover* dan Mutasi) yang bertujuan untuk mendapatkan kromosom yang lebih baik pada iterasi selanjutnya [7]. Pada penelitian ini kromosom merupakan urutan pekerjaan yang membentuk *string*. Sedangkan, gen merupakan pekerjaan yang terdapat di dalam kromosom.

Penggabungan Algoritma CE dan GA bertujuan untuk memperluas pencarian solusi pada sampel elit pada Algoritma CE dengan menambahkan mekanisme GA, yaitu *crossover* dan mutasi. Mekanisme GA berfungsi untuk menghindari pencarian solusi yang terjebak di area lokal optimal, karena *crossover* dan mutasi dapat menghasilkan kromosom baru yang memiliki sifat tetap menjaga kromosom pada populasi awalnya. Berikut tahapan-tahapan Algoritma CEGA [5]:

- 1). Inisialisasi parameter. Beberapa parameter yang akan ditentukan nilainya antara lain: Jumlah sampel (N) yang akan dibangkitkan. Parameter kejarangan (ρ), nilai ρ yang umum digunakan berkisar antara 1% – 10%. Koefisien penghalusan (α), nilai $0,4 < \alpha < 0,9$ merupakan nilai yang paling optimal [8]. Parameter crossover (P_c), memiliki nilai antara $0 \leq P_c \leq 1$. Parameter penghentian (β).
- 2.) Pembangkitan sampel. Sampel yang dibangkitkan pada iterasi pertama merupakan sampel yang dibangkitkan secara acak dengan membangkitkan bilangan acak sebanyak N . Kemudian, bilangan acak tersebut akan digunakan untuk menentukan pekerjaan yang akan dipilih. Sedangkan, sampel iterasi selanjutnya diperoleh dari mekanisme *crossover* dan mutasi. Jumlah sampel yang dibangkitkan sebanyak N sampel yaitu $X(1), \dots, X(N)$ yang telah ditentukan pada langkah 1. Sampel yang dibangkitkan merupakan urutan pekerjaan sepanjang N .
- 3). Perhitungan fungsi tujuan. Fungsi tujuan dihitung dengan meminimumkan nilai *makespan*. Nilai *makespan* diperoleh dari pekerjaan yang memiliki nilai *flow time* paling maksimum. *Flow time* (F_j) adalah rentang waktu pada saat pekerjaan siap untuk diproses sampai ketika pekerjaan selesai diproses. Jadi, *flow time* sama dengan penjumlahan waktu proses dengan waktu mulai produksi. Waktu proses ($WP_{i,j}$) adalah perkiraan waktu yang diperlukan untuk memproses suatu pekerjaan. Sedangkan, waktu mulai ($WM_{i,j}$) adalah waktu yang menunjukkan saat pekerjaan siap untuk dikerjakan [1]. Persamaan berikut un-

tuk menghitung nilai *makespan* pada tiap sampel yang dibangkitkan [5]:

$$Z = \max_{1 \leq j \leq n} (F_j) \quad (2.1)$$

dengan F_j adalah *flow time* pekerjaan urutan ke- j , *flow time* sama dengan waktu selesai pekerjaan pada mesin terakhir maka $F_j = WS_{m,j}$, dengan $j = 1, \dots, n$, m merupakan mesin ke- m dan $WS_{m,j} = WP_{m,j} + WM_{m,j}$. Nilai waktu mulai pada proses perhitungan fungsi tujuan menggunakan syarat sebagai berikut:

- i). Jika tidak terdapat pekerjaan pendahulu dan mesin pendahulu, maka waktu mulai pekerjaan = 0.

$$WM_{i,j} = 0 \quad (2.2)$$

- ii). Jika tidak terdapat pekerjaan pendahulu namun terdapat mesin pendahulu, maka waktu mulai pekerjaan = waktu selesai pekerjaan di mesin pendahulu.

$$WM_{i,j} = WS_{i-1,j} \quad (2.3)$$

- iii). Jika terdapat pekerjaan pendahulu namun tidak terdapat mesin pendahulu, maka waktu mulai pekerjaan = waktu selesai pekerjaan pendahulu.

$$WM_{i,j} = WS_{i,j-1} \quad (2.4)$$

- iv). Jika terdapat pekerjaan pendahulu dan terdapat mesin pendahulu, maka waktu mulai pekerjaan = waktu terlama di antara waktu selesai pekerjaan pendahulu dengan waktu selesai pekerjaan di mesin pendahulu.

$$WM_{i,j} = \max\{WS_{i,j-1}, WS_{i-1,j}\} \quad (2.5)$$

Persamaan berikut untuk menentukan waktu selesai pekerjaan ke- j pada mesin ke- i .

$$WS_{i,j} = WP_{i,j} + WM_{i,j} \text{ dengan } i = 1, \dots, m, j = 1, \dots, n.$$

Keterangan:

1. $WS_{i,j}$ adalah waktu selesai pekerjaan urutan ke- j pada mesin ke- i .
2. $WP_{i,j}$ adalah waktu proses pekerjaan urutan ke- j pada mesin ke- i .
3. $WM_{i,j}$ adalah waktu mulai pekerjaan urutan ke- j pada mesin ke- i .
- 4). Mengecek iterasi maksimal. Pengecekan iterasi maksimal menggunakan $\varepsilon = |P_c(k) - P_c(k-1)|$. Nilai $P_c(k)$ merupakan parameter *crossover* pada iterasi ke- k . Apabila nilai $\varepsilon > \beta$ maka kriteria pemberhentian belum tercapai dan lanjut kelangkah 5. Namun, apabila nilai $\varepsilon \leq \beta$ maka kriteria pemberhentian tercapai dan iterasi selesai.

- 5). Penentuan sampel elit. Jumlah sampel elit ditentukan oleh $\lceil \rho \times N \rceil$. Kemudian, diambil sampel elit dari sampel yang telah diurutkan yang memiliki nilai *makespan* terkecil sebanyak $\lceil \rho \times N \rceil$ [6].
- 6). Pembobotan sampel. Nilai pembobotan sampel elit diperoleh dari evaluasi terhadap sampel yang memiliki nilai *fitness* tertinggi pada iterasi sebelumnya. Bobot sampel elit yang diperoleh dijadikan sebagai dasar penentuan sampel yang dielitisme [6].
- 7). Perhitungan *linier fitness ranking* (LFR). Nilai LFR digunakan untuk pemilihan induk pada proses *crossover*. Persamaan berikut untuk menentukan nilai LFR,

$$\text{LFR}(i) = F_{maks} - (F_{maks} - F_{min}) \times \left(\frac{i - 1}{N - 1} \right) \quad (2.6)$$

dengan $i = 1, \dots, N$, $F_{maks} = 1/X(1)$, $F_{min} = 1/X(N)$ dan $\text{LFR}(i)$ adalah nilai LFR sampel ke- i .

- 8). Update parameter *crossover*. Proses ini dilakukan untuk mendapatkan nilai parameter baru. Parameter tersebut digunakan untuk evaluasi kriteria pemberhentian dan menentukan kromosom yang di *crossover*. Update parameter *crossover* dilakukan di setiap iterasi. Persamaan berikut untuk update parameter *crossover*,

$$P_c(k) = (1 - \alpha) \times u + (P_c(k - 1) \times \alpha),$$

dengan k merupakan iterasi ke- k dan $u = \frac{\bar{Z}_e}{2Z_{best}}$, $\bar{Z}_e$ adalah rata-rata *makespan* dari sampel elit dan Z_{best} adalah *makespan* terkecil tiap iterasi [6].

- 9). Elitisme bertujuan untuk mempertahankan sampel yang memiliki nilai fungsi tujuan terkecil atau nilai *fitness* tertinggi agar terpilih kembali pada iterasi berikutnya. Jumlah sampel yang di elitisme sebanyak jumlah sampel elit [6].
- 10). Penentuan pemilihan induk *crossover*. Proses ini dilakukan menggunakan mekanisme *roulette wheel*, yaitu berdasarkan nilai *fitness* setiap sampel. Prosedur pemilihan induk 1 dan induk 2 menggunakan *roulette wheel* adalah sebagai berikut:
 - a. Pemilihan induk 1, diambil dari sampel elit yang memiliki nilai fungsi tujuan terkecil dari mekanisme elitisme.
 - b. Pemilihan induk 2, dipilih berdasarkan nilai evaluasi dari sampel keseluruhan, yaitu menggunakan nilai LFR dari sampel yang sedang dievaluasi. Apabila nilai perbandingan kumulatif LFR dan total LFR lebih besar dari nilai acak yang dibangkitkan, maka sampel tersebut menjadi induk 2.
- 11). *Crossover* (pindah silang). Menurut [6] *crossover* adalah tahap pembentukan sampel baru (*offspring*) dari dua sampel induk yang terpilih (*parent*). Metode *crossover* yang digunakan adalah pindah silang dari dua titik (*2-point order crossover*), yaitu menukarkan gen (pekerjaan) diantara dua titik pada masing-masing induk. Persamaan berikut untuk menentukan pekerjaan dari sampel

induk yang akan di *crossover* adalah:

$$b_i = \lceil r_i \times n \rceil \text{ maka } p_i \text{ dengan } r_i \text{ adalah bilangan acak } [0, 1] \text{ untuk } i = 1, 2$$

Jika $b_1 < b_2$ maka b_1 menjadi p_1 dan b_2 menjadi p_2 , apabila $b_1 > b_2$ maka b_1 menjadi p_2 dan b_2 menjadi p_1 .

- 12). Mutasi, bertujuan untuk menghasilkan sampel baru (*offspring*) dengan cara menukarkan beberapa pekerjaan pada suatu sampel. Parameter mutasi (P_m) digunakan untuk menentukan jumlah sampel yang mengalami mutasi [7]. Nilai parameter mutasi ditentukan dengan rumus sebagai berikut [5],

$$P_m = \frac{P_c}{2},$$

Jumlah sampel yang dimutasi adalah $n = \lceil P_m \times N \rceil$. Persamaan untuk menentukan bagian sampel yang dimutasi (I, J), yaitu:

$$I = \lceil r_1 \times n \rceil \text{ dan } J = \lceil r_2 \times n \rceil.$$

Persamaan berikut untuk menentukan mekanisme mutasi (K) yaitu:

$$K = \lceil r \times 3 \rceil,$$

dengan r , r_1 , dan r_2 adalah bilangan acak $[0, 1]$. Apabila nilai $K = 1$ maka dilakukan *flip mutation* (membalik), jika $K = 2$ dilakukan *swap mutation* (menukar), dan jika $K = 3$ maka dilakukan *slide mutation* (menggeser). Setelah diperoleh sampel yang baru dari hasil mutasi maka kembali ke langkah 3 untuk melakukan perhitungan fungsi tujuan yang baru.

3. Pembahasan

Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari [4]. Data tersebut merupakan data waktu standar proses lamanya pengerjaan bahan baku batu alam pada tiap mesin di perusahaan CV Bestone Indonesia. Adapun data waktu lamanya proses pekerjaan pada tiap mesin dapat dilihat pada Tabel 1 berikut.

Tabel 1. Data waktu lamanya proses pekerjaan pada tiap mesin

No.	Jenis Produk	Waktu Proses (detik)				
		M1	M2	M3	M4	M5
1	BEST 20 SH	74,66	86400	221,88	43200	22,66
2	BEST 47 A	61,71	86400	166,38	43200	23,25
3	BEST 47	62,62	86400	166,04	43200	25,22
4	BEST 50 SH	64,42	86400	176,45	43200	23,99

Pekerjaan 1 didefinisikan sebagai proses pengerjaan produk BEST 20 SH, pekerjaan 2 didefinisikan sebagai proses pengerjaan produk BEST 47 A, pekerjaan 3 didefinisikan sebagai proses pengerjaan produk BEST 47 dan pekerjaan 4 didefinisikan sebagai proses pengerjaan produk BEST 50 SH. Setiap pekerjaan dikerjakan

diseluruh mesin yaitu pada proses pemotongan sebagai M1, proses pengeringan pertama sebagai M2, proses perakitan sebagai M3, proses pengeringan kedua sebagai M4 dan proses penyimpanan sebagai M5.

Tahapan Algoritma CEGA pada penjadwalan *flow shop* adalah inisialisasi parameter, yaitu jumlah sampel (N) sebanyak 6, parameter kejaran (ρ) sebesar 0,01, koefisien penghalusan (α) sebesar 0,6, parameter *crossover* (P_c) sebesar 1, dan parameter pemberhentian (β) sebesar 0,0000001. Tahap selanjutnya adalah membangkitkan sampel awal dengan membangkitkan bilangan acak diantara 0 dan 1 sebanyak 4. Berikut bilangan acak yang digunakan untuk membangkitkan urutan pekerjaan sampel $X(1)$ pada Tabel 2,

Tabel 2. Bilangan acak pada sampel $X(1)$

Pekerjaan	Bilangan Acak	No. Urut
1	0,6723	3
2	0,7320	4
3	0,2669	1
4	0,4758	2

Bilangan acak pada Tabel 2 diurutkan berdasarkan nilai terkecil. Bilangan acak terkecil adalah 0,2669 berada di pekerjaan 3, maka pekerjaan 3 menjadi pekerjaan urutan pertama. Proses ini berlanjut sampai pada bilangan acak terbesar yaitu 0,7320 berada di pekerjaan 2, sehingga pekerjaan 2 menjadi pekerjaan urutan ke-4. Urutan pekerjaan yang didapat dari proses tersebut yaitu $X(1) = 3 - 4 - 1 - 2$. Diperoleh sampel awal yaitu:

$$\begin{array}{lll} X(1) = 3 - 4 - 1 - 2 & X(3) = 1 - 3 - 4 - 2 & X(5) = 4 - 2 - 1 - 3 \\ X(2) = 1 - 2 - 4 - 3 & X(4) = 2 - 1 - 3 - 4 & X(6) = 2 - 3 - 1 - 4 \end{array}$$

Selanjutnya dilakukan penghitungan fungsi tujuan setiap sampel dengan menggunakan Persamaan (2.1). Berikut perhitungan nilai *flow time* $X(1)$ pada pekerjaan urutan ke-1: $F_1 = WS_{5,1}$, pekerjaan urutan ke-1 adalah pekerjaan 3 Berdasarkan Persamaan (2.2) maka nilai ($WM_{1,1} = 0$), karena pekerjaan urutan ke-1 pada mesin ke-1 tidak ada pekerjaan pendahulu dan tidak ada mesin pendahulu.

$$WS_{1,1} = WP_{1,1} + WM_{1,1} = 62, 62 + 0 = 62, 62.$$

Berdasarkan Persamaan (2.3) maka nilai ($WM_{2,1} = WS_{1,1} = 62, 62$), karena pekerjaan urutan ke-1 pada mesin ke-2 memiliki mesin pendahulu yaitu mesin ke-1, namun tidak memiliki pekerjaan pendahulu.

$$WS_{2,1} = WP_{2,1} + WM_{2,1} = 86400 + 62, 62 = 86462, 62.$$

Berdasarkan Persamaan (2.3) maka nilai ($WM_{3,1} = WS_{2,1} = 86462, 62$), karena pekerjaan urutan ke-1 pada mesin ke-3 memiliki mesin pendahulu yaitu mesin ke-2, namun tidak memiliki pekerjaan pendahulu.

$$WS_{3,1} = WP_{3,1} + WM_{3,1} = 166, 04 + 86462, 62 = 86628, 66.$$

Berdasarkan Persamaan (2.3) maka nilai $(WM_{4,1}) = WS_{3,1} = 86628,66$, karena pekerjaan urutan ke-1 pada mesin ke-4 memiliki mesin pendahulu yaitu mesin ke-3, namun tidak memiliki pekerjaan pendahulu.

$$WS_{4,1} = WP_{4,1} + WM_{4,1} = 43200 + 86628,66 = 129828,66.$$

Berdasarkan Persamaan (2.3) maka nilai $(WM_{5,1} = WS_{4,1}) = 129828,66$, karena pekerjaan urutan ke-1 pada mesin ke-5 memiliki mesin pendahulu yaitu mesin ke-4, namun tidak memiliki pekerjaan pendahulu.

$$WS_{5,1} = WP_{5,1} + WM_{5,1} = 25,22 + 129828,66 = 129853,88.$$

Jadi, diperoleh nilai F_1 sebesar 129853,88. Perhitungan *flow time* untuk pekerjaan urutan selanjutnya menggunakan proses yang sama pada pekerjaan urutan ke-1. Sehingga diperoleh *flow time* setiap pekerjaan yaitu $F_1 = 129853,88$; $F_2 = 216263,06$; $F_3 = 302707,16$ dan $F_4 = 389052,52$. Berdasarkan nilai *flow time* maka nilai makespan kromosom $X(1)$ adalah

$$Z = \max\{F_1, F_2, F_3, F_4\} = 389052,52$$

untuk menghitung makespan sampel lainnya, menggunakan langkah yang sama dengan sampel $X(1)$. Sehingga diperoleh *makespan* tiap sampel pada Tabel 3 berikut,

Tabel 3. Nilai *makespan* untuk 6 sampel

Sampel	Makespan	Sampel	Makespan
$X(1)$	389052,25	$X(4)$	389062,15
$X(2)$	389065,92	$X(5)$	389055,68
$X(3)$	389064,29	$X(6)$	890620,15

kemudian sampel pada tabel 3 diurutkan dari yang terkecil hingga terbesar berdasarkan nilai *makespan*. Sehingga, sampel $X(1)$ tetap karena memiliki nilai *makespan* terkecil, sampel $X(5)$ akan menjadi sampel $X(2)$ dengan urutan pekerjaan $4-2-1-3$, sampel $X(4)$ akan menjadi sampel $X(3)$ dengan urutan pekerjaan $2-1-3-4$, sampel $X(6)$ akan menjadi sampel $X(4)$ dengan urutan pekerjaan $2-3-1-4$, sampel $X(3)$ akan menjadi sampel $X(5)$ dengan urutan pekerjaan $1-3-4-2$ dan sampel $X(2)$ akan menjadi sampel $X(6)$ dengan urutan pekerjaan $1-2-4-3$. Pengurutan sampel ini akan digunakan untuk tahap selanjutnya.

Nilai ρ telah ditentukan sebesar 0,01 maka jumlah sampel elit adalah $\lceil \rho \times N \rceil = \lceil 0,01 \times 6 \rceil = 1$. Kemudian, diambil satu sampel elit dari sampel yang telah diurutkan yang memiliki nilai *makespan* paling kecil. Sehingga, diperoleh sampel elit yaitu sampel $X(1)$. Karena proses ini merupakan iterasi pertama dan hanya terdapat 1 sampel elit, maka bobot sampel elit tersebut sebesar 1. Selanjutnya, dilakukan perhitungan nilai LFR. Berikut perhitungan nilai LFR menggunakan Persamaan (2.6),

$$Fmax = 1/X(1) = 0,00000257035 \text{ dan } Fmin = 1/X(6) = 0,00000257026.$$

Maka nilai LFR untuk ke 6 sampel adalah LFR (1) = 0,00000257035, LFR (2) = 0,00000257033, LFR (3) = 0,00000257031, LFR (4) = 0,00000257029, LFR (5) = 0,00000257028, dan LFR (6) = 0,00000257026.

Proses *update* parameter *crossover* dilakukan dengan menghitung nilai $u = 389052,52 / (2 \times 389052,52) = 0.5$ maka diperoleh $P_c(2) = 0,8$. Jumlah sampel yang di elitisme sebanyak jumlah sampel elit. Karena pembobotan sampel elit bernilai 1. Maka, jumlah sampel yang di elitisme adalah 1, berdasarkan nilai makespan sampel yang paling minimum yaitu sampel $X(1)$. Sehingga sampel $X(1)$ terpilih sebagai sampel yang di elitisme. Pemilihan induk *crossover* dilakukan dengan menggunakan mekanisme *roulette wheel* berdasarkan nilai *fitness* sampel. Pemilihan induk 1 diambil dari sampel elit yang memiliki nilai fungsi tujuan terkecil, berdasarkan proses elitisme sampel $X(1) = 3-4-1-2$ terpilih sebagai induk 1. Pemilihan induk 2 menggunakan nilai LFR dari sampel yang sedang dievaluasi. Apabila nilai perbandingan kumulatif LFR dan total LFR lebih besar dari bilangan acak yang dibangkitkan, maka sampel tersebut menjadi induk 2. Total LFR adalah 0,00001542182.

Kumulatif LFR total dari setiap sampel, yaitu $X(1) = 0,166669595$, $X(2) = 0,333338018$, $X(3) = 0,500000527$, $X(4) = 0,666671351$, $X(5) = 0,833336261$, dan $X(6) = 1$. Selanjutnya bilangan acak yang dibangkitkan adalah 0,4851, dengan demikian nilai kumulatif LFR total untuk sampel $X(1)$ dan $X(2)$ kurang dari bilangan acak yang dibangkitkan. Sedangkan, untuk sampel $X(3)$, $X(4)$, $X(5)$ dan $X(6)$ memiliki nilai kumulatif LFR total yang lebih besar dari bilangan acak yang dibangkitkan. Sehingga, diperoleh sampel $X(1)$ sebagai induk 1 dari sampel elit karena memiliki nilai fungsi tujuan terbaik dari mekanisme elitisme. Sedangkan, sampel $X(3)$, $X(4)$, $X(5)$ dan $X(6)$ terpilih sebagai induk 2.

Proses *crossover* menggunakan metode *2-point order crossover*. Untuk mengetahui sampel yang mengalami *crossover* maka dibangkitkan bilangan acak (R). Jika $R < P_c$ maka dilakukan *crossover*. Sedangkan, jika $R > P_c$ maka tidak dilakukan *crossover*. bilangan acak (R) yang dibangkitkan disajikan dalam Tabel 4 berikut,

Tabel 4. Pembangkitan bilangan R penentuan *crossover*

Sampel	R	Sampel	R
$X(1)$	0,86113	$X(4)$	0,54866
$X(2)$	0,58979	$X(5)$	0,47786
$X(3)$	0,11668	$X(6)$	0,59866

Proses *crossover* dilakukan pada sampel $X(3)$, $X(4)$, $X(5)$, dan $X(6)$, karena nilai R dari sampel tersebut lebih kecil dari $P_c = 0,8$. Sedangkan, sampel $X(2)$ tidak dilakukan *crossover* karena tidak terpilih sebagai induk 2. Selanjutnya untuk menentukan bagian dari sampel induk yang akan di *crossover*, yaitu dengan membangkitkan dua bilangan acak. Berikut *crossover* untuk sampel $X(1)$ dengan sampel $X(3)$. Diperoleh dua bilangan acak $r_1 = 0,4899$ dan bilangan $r_2 = 0,6789$, maka $b_1 = \lceil 0,4899 \times 4 \rceil = \lceil 1,9596 \rceil = 2$ dan $b_2 = \lceil 0,6789 \times 4 \rceil = \lceil 2,7156 \rceil = 3$ sehingga $p_1 = 2$ dan $p_2 = 3$.

sampel induk $X(1) = 3 - 4 - 1 - 2$
sampel induk $X(3) = 1 - 2 - 4 - 3$

Selanjutnya pekerjaan yang terpilih tersebut ditukar antar sampel induk seperti berikut:

sampel baru 1 = $\dots - \dots - \mathbf{4} - \dots$
sampel baru 2 = $\dots - \dots - \mathbf{1} - \dots$

untuk mengisi titik-titik pada sampel baru 1, maka diperiksa satu persatu pekerjaan pada sampel $X(1)$. Jika terdapat pekerjaan pada sampel $X(1)$ yang belum tercantum pada sampel baru 1, maka pekerjaan tersebut akan mengisi titik-titik yang kosong. Hal tersebut dilakukan secara berurutan, sehingga sampel hasil crossover menjadi:

sampel baru 1 = $3 - 1 - 4 - 2$ menggantikan sampel $X(2)$ yang lama
sampel baru 2 = $4 - 3 - 1 - 2$ menggantikan sampel $X(3)$ yang lama.

Kemudian evaluasi induk 2 lainnya dengan proses yang sama, yang memiliki nilai bilangan acak (R) lebih kecil dari P_c . Setelah proses *crossover* dilakukan, diperoleh sampel berikut,

$X(1) = 3 - 4 - 1 - 2$ $X(3) = 4 - 3 - 1 - 2$ $X(5) = 4 - 2 - 1 - 3$
 $X(2) = 3 - 1 - 4 - 2$ $X(4) = 3 - 2 - 1 - 4$ $X(6) = 3 - 2 - 1 - 4$

Proses mutasi dipilih dengan cara membangkitkan bilangan acak dan pekerjaan pada sampel akan ditukar dengan pekerjaan lainnya. Nilai parameter mutasi ditentukan dengan rumus sebagai berikut $P_m = P_c/2 = 0,8/2 = 0,4$. Jumlah sampel yang akan dimutasi adalah $Na = \lceil P_m \times N \rceil = \lceil 0,4 \times 6 \rceil = \lceil 2,4 \rceil = 3$, sehingga ada 3 sampel yang mengalami proses mutasi. Untuk mengetahui sampel mana yang mengalami mutasi maka akan dibangkitkan bilangan acak (R). Jika $R < P_m$ maka dilakukan proses mutasi, dan jika $R > P_m$ maka tidak dilakukan proses mutasi. Berikut sampel yang dimutasi pada Tabel 5,

Tabel 5. Penentuan sampel yang dimutasi

Sampel	R	P_m	Keterangan	Sampel	R	P_m	Keterangan
$X(1)$	0,1442		tetap	$X(4)$	0,2669	0,4	dimutasi
$X(2)$	0,1442	0,4	dimutasi	$X(5)$	0,3851	0,4	dimutasi
$X(3)$	0,4591	0,4	tidak dimutasi	$X(6)$	0,4782	0,4	tidak dimutasi

Selanjutnya melakukan proses mutasi pada sampel yang terpilih. Berikut ini diberikan proses mutasi pada sampel $X(2)$. Diketahui $n = 4$ dan dua bilangan acak yang dibangkitkan $r_1 = 0,1979$ dan $r_2 = 0,9461$, maka $I = \lceil 4 \times 0,1979 \rceil = \lceil 0,7916 \rceil = 1$ dan $J = \lceil 4 \times 0,9461 \rceil = \lceil 3,7844 \rceil = 4$. Kemudian membangkitkan nilai r secara acak pada selang $[0,1]$ dan diperoleh nilai $r = 0,4132$. Sehingga nilai $K = \lceil 0,4132 \times 3 \rceil = \lceil 1,2396 \rceil = 2$. Karena $K = 2$, maka dilakukan pembalikan pada sampel $X(2) = 3 - 1 - 4 - 2$, sehingga diperoleh sampel $X(2)$ yang baru, yaitu

3 – 2 – 4 – 1. Proses mutasi pada sampel $X(4)$ dan sampel $X(5)$, dilakukan dengan langkah yang sama seperti proses mutasi sampel $X(2)$. Berikut sampel baru hasil mutasi, yaitu

$$\begin{array}{lll} X(1) = 3 - 4 - 1 - 2 & X(3) = 4 - 3 - 1 - 2 & X(5) = 4 - 1 - 2 - 3 \\ X(2) = 3 - 2 - 4 - 1 & X(4) = 3 - 1 - 2 - 4 & X(6) = 3 - 2 - 1 - 4 \end{array}$$

Setelah itu menghitung nilai *makespan* (fungsi tujuan) dari sampel tersebut. Nilai *makespan* dari sampel yang baru disajikan pada Tabel 6.

Tabel 6. Nilai fungsi tujuan (*makespan*) sampel baru

Sampel	<i>Makespan</i>	Sampel	<i>Makespan</i>
$X(1)$	389052,25	$X(4)$	389063,06
$X(2)$	389107,16	$X(5)$	389055,68
$X(3)$	389054,05	$X(6)$	389063,06

berdasarkan Tabel 6 diperoleh nilai *makespan* terkecil pada iterasi pertama yaitu sebesar 389052,25. Proses iterasi dilakukan sampai kriteria pemberhentian terpenuhi. kriteria pemberhentian terpenuhi pada saat iterasi ke-31 dengan nilai ε sebesar 0,00000007369, sedangkan nilai β sebesar 0,0000001 sehingga $\varepsilon < \beta$ dan proses iterasi dihentikan. Pada Tabel 7 dapat dilihat hasil percobaan sebanyak 16 kali, dimana masing-masing penghitungan dilakukan sampai kriteria pemberhentian terpenuhi,

Tabel 7. Hasil urutan pekerjaan dengan Algoritma CEGA

No	Urutan pekerjaan	<i>Makespan</i>
1	3 – 4 – 1 – 2	389052,25
2	3 – 4 – 1 – 2	389052,25
3	3 – 4 – 1 – 2	389052,25
4	3 – 1 – 4 – 2	389052,25
5	3 – 4 – 1 – 2	389052,25
6	3 – 4 – 1 – 2	389052,25
7	2 – 1 – 4 – 3	389052,97
8	3 – 4 – 1 – 2	389052,25
9	3 – 4 – 1 – 2	389052,25
10	3 – 4 – 1 – 2	389052,25
11	3 – 1 – 4 – 2	389052,25
12	3 – 4 – 1 – 2	389052,25
13	3 – 4 – 1 – 2	389052,25
14	3 – 1 – 4 – 2	389052,25
15	3 – 4 – 1 – 2	389052,25
16	3 – 4 – 1 – 2	389052,25

berdasarkan Tabel 7, diperoleh urutan pekerjaan yang memiliki fungsi tujuan terkecil yaitu urutan pekerjaan 3–4–1–2 dengan urutan pengerjaan produk BEST 47, produk BEST 20 SH, produk BEST 50 SH, produk BEST 47 A atau urutan pekerjaan 3–1–4–2 dengan urutan pengerjaan produk BEST 47, produk BEST 50 SH, produk BEST 20 SH, produk BEST 47 A dengan nilai *makespan* sebesar 389052,25 detik.

4. Kesimpulan

Berdasarkan pembahasan yang telah dilakukan, permasalahan penjadwalan *flow shop* dapat diselesaikan dengan Algoritma CEGA. Pada penyelesaian permasalahan penjadwalan *flow shop* di CV. Bestone Indonesia menggunakan Algoritma CEGA dengan nilai parameter yang digunakan yaitu N sebanyak 6, ρ sebesar 0,01, P_{ps} sebesar 1, α sebesar 0,6 dan β sebesar 0,000001 menghasilkan urutan pengerjaan produk BEST 47, produk BEST 20 SH, produk BEST 50 SH, produk BEST 47 A dan urutan pengerjaan produk BEST 47, produk BEST 50 SH, produk BEST 20 SH, produk BEST 47 A. Nilai *makespan* paling minimum yang diperoleh yaitu sebesar 389052,25 detik atau 108,09 jam dengan kriteria pemberhentian ε sebesar 0,00000007369 .

Daftar Pustaka

- [1] Ginting, R., 2009, *Penjadwalan Mesin*, Graha Ilmu, Yogyakarta
- [2] Haming, M. dan Nurnajamuddin, M., 2014, *Manajemen Produksi Modern Operasi Manufaktur dan Jasa, Buku 2*, Cahaya Prima Sentosa, Jakarta
- [3] Arkeman, Y., Seminar, B., dan Gunawan, H., 2012, *Algoritma Genetika, Teori dan Aplikasinya untuk Bisnis dan Industri*, IPB Press, Bogor
- [4] Kurniawati, A. dan Karim, S., 2016, Penjadwalan Produksi Flow Shop dengan Metode Ignal-Scharge dan Algoritma Nawaz, Ensore dan Ham (NEH) di CV. Bestone Indonesia, *Jurnal Sains, Teknologi dan Industri* Vol. **13**: 229 – 241
- [5] Widodo, S., Santoso, B. dan Siswanto, E., 2014. Pendekatan Algoritma Cross EntropyGenetic Algorithm untuk Menurunkan Makespan Pada Penjadwalan Flow Shop, *JEMIS* Vol. **2**: 41 – 49
- [6] Nurkhalida, L. dan Santosa, B., 2012, *Pendekatan Algoritma Cross EntropyGenetic Algorithm Pada Permasalahan Multi Objective Job Shop Scheduling*, (serial online) Available at: <https://www.researchgate.net/publication/267976436-pendekatan-cross-entropy-genetic-algorithmn-pada-permasalahan-multi-objective-job-shop-scheduling>. Pdf, (14 Mei 2021)
- [7] Syarif, A., 2014, *Algoritma Genetika Teori dan Aplikasi*, Edisi ke-2, Graha Ilmu, Yogyakarta
- [8] Boer, D., Kroese, D.P., Mannor, S., and Rubinstein, R.Y., 2005, A tutorial on the Cross-Entropy Method, *Annals of Operations Research* Vol. **134**: 19 – 67
- [9] Bashori, H.,Pratikto, dan Sugiono, 2015, Penjadwalan *Flow Shop* dengan Penerapan Algoritma *Cross Entropy-Genetic Algorithm* (CEGA) untuk Meminimasi *Makespan*, *JEMIS* Vol. **3**: 35 – 42
- [10] Muharni, Y., Saeful M, A. I., dan Rubyanti, T. E., 2019, Penjadwalan *Flow Shop* Mesin Paralel Menggunakan Metode *Longest Processing Time* dan *Cross*

Entropy-Genetic Algorithm Pada Pembuatan Produk Steel Bridge B-60, *Jurnal Ilmiah Teknik Industri* Vol. **7**: 213 – 225

- [11] Bahsori, H, 2015, Upaya Meminimasi *Makespan* dengan Penerapan Algoritma *Cross Entropy* Pada Penjadwalan *Flow Shop*, *Widya Teknika* Vol. **23**: 10 – 14
- [12] Alfandianto, A., Nugroho, Y. A., dan Setiafindari, W., 2017, Penjadwalan Produksi Menggunakan Pendekatan Algoritma Genetika di PT Pertani (PERSERO) Cabang D.I Yogyakarta, *Jurnal DISPROTEK* Vol. **2**: 1 – 7
- [13] Suprayogi, D. A., dan Mahmudy, W. F., 2014, Penerapan Algoritma Genetika *Traveling Salesman Problem with Time Window*, *Jurnal Buana Informatika* Vol. **6**: 121 – 130
- [14] Utama, D. M., Ardiansyah, L. R., dan Garside, A. K., 2019, Penjadwalan *Flow Shop* untuk Meminimasi *Total Tardiness* Menggunakan Algoritma *Cross Entropy-Genetic Algorithmn*, *Jurnal Optimasi Sistem Industri* Vol. **18**: 133 – 141
- [15] Saputro, N., dan Yento, 2004, Pemakaian Algoritma Genetik untuk Penjadwalan *Jop Shop Dinamis Non Deterministik*, *Jurnal Teknik Industri* Vol. **6**: 61 – 70