

## Perbandingan Algoritma Penjadwalan *Symbiotic Organisms Search* dan *Earliest Due Date*: Studi Kasus di PT Citra Indah Abadi Jaya

Davin Hariyanto<sup>1</sup>, Tanti Octavia<sup>2</sup>, Iwan Halim Sahputra<sup>3</sup>

**Abstract:** PT Citra Indah Abadi Jaya is a service company engaged in the powder *coatings* of aluminum and metal. The problem that occurs in companies is that there is *lateness* in order completion. This *lateness* was caused by company scheduling based on the estimation of the Production Department. Therefore, a production scheduling algorithm is needed that can minimize the maximum *lateness* in the company. The algorithm that will be used in this research is the *Symbiotic Organisms Search* (SOS) and *Earliest Due Date* (EDD) algorithms. The SOS and EDD algorithm programs are run with Excel VBA. The eco size of the SOS algorithm is 50 with 20 replications. The results of the proposed best algorithm will be compared with scheduling the company's actual conditions. It is expected that the proposed scheduling algorithm can meet the needs of the company.

**Keywords:** production scheduling; SOS algorithm; EDD algorithm; minimizing maximum *lateness*

### Pendahuluan

PT Citra Indah Abadi Jaya merupakan perusahaan jasa yang bergerak dalam bidang pengecatan *powder coating* aluminium dan logam. *Powder coating* adalah proses pelapisan pewarnaan dengan bahan serbuk pada permukaan logam termasuk aluminium dan besi. Proses produksi PT Citra Indah Abadi Jaya meliputi proses *treatment*, *spraying*, *curing* (oven), dan *packing*. Permasalahan yang terjadi pada PT Citra Indah Abadi Jaya ialah terdapat beberapa pekerjaan yang mengalami keterlambatan penyelesaian pesanan. Data pesanan perusahaan periode 2019 menunjukkan bahwa 42% dari total pekerjaan mengalami keterlambatan. Keterlambatan penyelesaian pesanan ini disebabkan penjadwalan yang dilakukan perusahaan berdasarkan estimasi dari pengalaman kerja yang dilakukan oleh departemen produksi. Pesanan yang dianggap prioritas bagi perusahaan selalu dijadwalkan mendahului pesanan umum. Estimasi penjadwalan perusahaan seringkali menghasilkan *earliness* bagi pesanan prioritas. Kondisi ini menghasilkan beberapa keterlambatan pada pesanan umum lainnya. Perusahaan memiliki target untuk menghasilkan maksimum keterlambatan penyelesaian pesanan yang minimal. Tujuan adanya

target ini ialah untuk memenuhi kepuasan pelanggan. Oleh karena itu, perusahaan membutuhkan penjadwalan yang dapat meminimalkan maksimum keterlambatan. Karakteristik permasalahan dari perusahaan adalah minimum maksimum *lateness* untuk *single machine processing* dengan jumlah  $n$  *job*. Usulan dalam penelitian ini ialah algoritma penjadwalan produksi perusahaan yang dapat meminimalkan maksimum keterlambatan perusahaan. Penelitian ini menggunakan algoritma *Symbiotic Organisms Search* (SOS) dan algoritma *Earliest Due Date* (EDD) sebagai usulan penjadwalan produksi. Penelitian ini ingin mencoba membandingkan performansi algoritma SOS dan EDD dalam meminimumkan maksimum *lateness*.

### Metode Penelitian

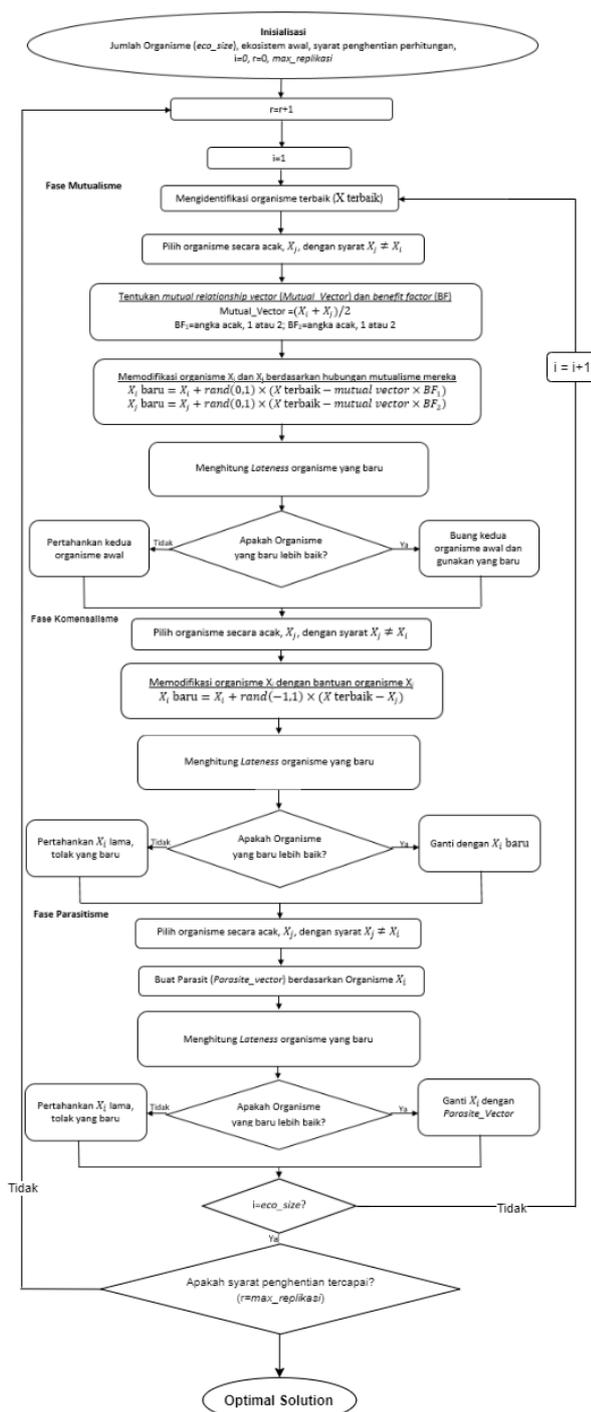
Metode yang akan digunakan untuk menyelesaikan permasalahan dalam penelitian ini yaitu algoritma SOS dan EDD:

#### *Symbiotic Organisms Search*

Algoritma metaheuristik SOS merupakan algoritma yang menyimulasikan interaksi-interaksi simbiosis oleh sepasang organisme dalam suatu ekosistem. Interaksi antar organisme yang disimulasikan dalam algoritma ini ialah simbiosis mutualisme, simbiosis komensalisme, dan simbiosis parasitisme. Algoritma

<sup>1,2,3</sup> Fakultas Teknologi Industri, Jurusan Teknik Industri, Universitas Kristen Petra. Jl. Siwalankerto 121-131, Surabaya 60236. Email: [davin\\_hariyanto@ymail.com](mailto:davin_hariyanto@ymail.com), [tanti@petra.ac.id](mailto:tanti@petra.ac.id), [iwanh@petra.ac.id](mailto:iwanh@petra.ac.id)

SOS dapat digambarkan melalui *flowchart* seperti pada Gambar 1 (Cheng *et al.* [1]).



Gambar 1. Alur SOS (Cheng [1])

Simbiosis mutualisme adalah interaksi organisme dengan kondisi masing-masing organisme sebagian atau seluruhnya saling diuntungkan satu dengan yang lain. Tahap mutualisme membutuhkan variabel solusi baru, yakni  $X_i$  baru,  $X_j$  baru dan *mutual vector* (MV) melalui Persamaan 1 hingga 3. Variabel solusi baru yang dihasilkan dibandingkan

dengan  $X_i$  dan  $X_j$  untuk mendapatkan organisme terbaik.

$$X_i \text{ baru} = X_i + \text{rnd}(0,1) \times (X_{\text{terbaik}} - MV \times BF_1) \quad (1)$$

$$X_j \text{ baru} = X_j + \text{rnd}(0,1) \times (X_{\text{terbaik}} - MV \times BF_2) \quad (2)$$

$$MV = (X_i + X_j)/2 \quad (3)$$

Keterangan:

$X_i$  = Organisme ke- $i$  dari ekosistem

$X_j$  = Organisme ke- $j$  dari ekosistem

$j$  = Nilai acak dari 1 hingga nilai *ecosize*

$BF_1$  = nilai acak, 1 atau 2

$BF_2$  = nilai acak, 1 atau 2

Simbiosis komensalisme adalah interaksi organisme dengan kondisi salah satu organisme diuntungkan dan yang lain tidak dirugikan maupun diuntungkan. Tahap komensalisme membutuhkan variabel solusi baru melalui Persamaan 4. Variabel yang dihasilkan dibandingkan dengan  $X_i$  untuk mendapatkan organisme terbaik.

$$X_i \text{ baru} = X_i + \text{rand}(-1,1) \times (X_{\text{terbaik}} - X_j) \quad (4)$$

Simbiosis parasitisme adalah interaksi organisme dengan kondisi salah satu organisme diuntungkan dan yang lain dirugikan. Tahap parasitisme membutuhkan variabel solusi baru yang diberi nama *parasite vector*. *Parasite vector* dihasilkan melalui hasil mutasi  $X_i$  dengan variabel acak. Variabel yang dihasilkan dibandingkan dengan  $X_j$  untuk mendapatkan organisme terbaik.

### Earliest Due Date

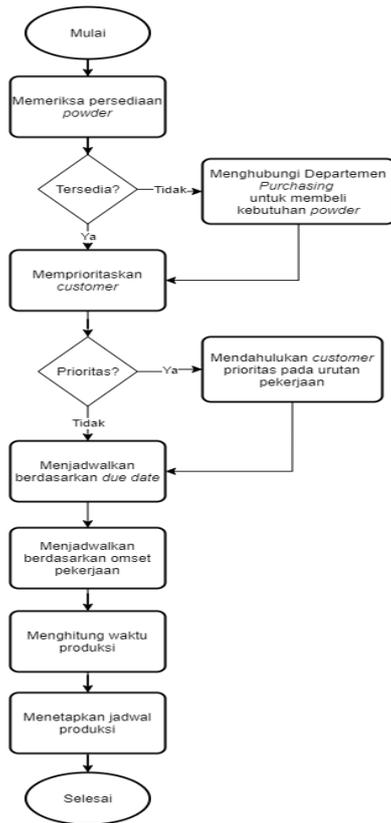
Urutan pekerjaan dari hasil metode penjadwalan EDD memperhatikan waktu jatuh tempo terdekat atau perlu didahulukan. Parameter yang digunakan dalam metode penjadwalan EDD adalah batas waktu yang ada dan waktu pengerjaan atau proses dari pekerjaan. Metode ini dapat digunakan untuk menjadwalkan pekerjaan *single machine* dan *parallel machine*. Hasil dari penjadwalan dengan metode ini ialah minimum nilai keterlambatan maksimum (Bedworth *et al.* [2]).

## Hasil dan Pembahasan

### Penjadwalan Perusahaan

Selama ini penjadwalan proses produksi perusahaan dilakukan oleh Departemen Produksi. Jam kerja

dalam satu hari ialah delapan jam dengan lima hari kerja dalam setiap minggu (Senin hingga Jumat). Departemen Produksi melakukan penjadwalan produksi setiap hari Jumat untuk periode satu minggu berikutnya. Penjadwalan produksi dapat dilakukan di luar hari Jumat ketika terdapat pesanan mendesak. Pesanan mendesak yang dimaksud merupakan permintaan dari pelanggan prioritas. Alur penjadwalan produksi perusahaan dapat dilihat pada Gambar 2.



**Gambar 2.** Alur penjadwalan produksi perusahaan

Alur dari penjadwalan produksi perusahaan diawali dengan departemen produksi menerima surat jalan. Departemen produksi akan mengkonfirmasi ketersediaan bubuk *coating* dengan departemen gudang. Departemen produksi segera menghubungi departemen pengadaan untuk memesan bubuk *coating* bila kekurangan. Setelah kesiapan *material* dikonfirmasi, departemen produksi mulai mengurutkan pekerjaan.

Pengurutan pekerjaan awal ialah berdasarkan *due date*. *Due date* pekerjaan akan diurutkan dari yang terdekat. Apabila ditemukan dua atau lebih *due date* pekerjaan yang sama, maka akan diurutkan berdasarkan prioritas pelanggan. Prioritas pelanggan dibagi menjadi dua, yakni pelanggan umum dan pelanggan prioritas. Ketika ditemukan dua atau lebih perbandingan pekerjaan dengan prioritas yang sama, maka akan diurutkan

berdasarkan omset pekerjaan. Omset pekerjaan diurutkan mulai dari omset terbesar hingga terkecil.

### Usulan Algoritma Penjadwalan

Algoritma penjadwalan yang diusulkan akan menentukan urutan pengerjaan pesanan. Urutan pekerjaan yang dihasilkan akan memberikan maksimal keterlambatan yang terkecil. Waktu proses merupakan total waktu dari tahap *treatment* hingga *packing* selesai. Waktu proses setiap pekerjaan bervariasi karena mengikuti karakteristik dari pesannya. Karakteristik pesanan yang dimaksud meliputi deskripsi pekerjaan, kategori *customer*, tanggal penerimaan barang *customer*, dan informasi setiap tahapan proses. Waktu total dari tahap *treatment* dipengaruhi oleh jumlah keranjang dan jenis *treatment*. Kondisi dari satu kali rotasi akan menjalankan maksimal dua keranjang. Perbedaan jenis *treatment* mempengaruhi lama waktu setiap rotasi, yang dibedakan menjadi tiga yaitu MF, CA, dan Plat.

Waktu total dari tahap proses dipengaruhi jumlah batang atau lembar setiap pekerjaan, jumlah lembar atau batang di setiap *hollow* dan karakteristik warna. Waktu proses dipengaruhi jumlah kebutuhan *hollow* setiap pekerjaan. Faktor lain yang mempengaruhi waktu proses adalah karakteristik warna terkait pergantian karakteristik warna bubuk *coating*. Kode dari karakteristik warna pada algoritma ialah “0” dan “1” untuk warna gelap dan terang. Waktu *packing* dipengaruhi jumlah batang dan jenis *packing*. Material plat dan profil akan membutuhkan waktu *unloading* 10 menit dan 20 menit, karena terdapat waktu menunggu turunnya suhu material yang berbeda. Faktor lainnya ialah perbedaan jenis *packing* dan permintaan *protect material* dari pelanggan. Jenis *packing* dibedakan menjadi empat, yakni profil, profil dengan *protect* (profil p), plat dan plat dengan *protect* (plat p).

Penelitian ini memiliki fungsi tujuan dan variabel keputusan dari permasalahan yang ada. Variabel keputusan dari permasalahan ialah karakteristik dari pesanan. Fungsi tujuan dari permasalahan ialah maksimum keterlambatan terkecil. Berikut fungsi matematis dari permasalahan yang ada:

$$Z = \text{Min} (\text{Max } L_k) \tag{5}$$

Keterangan:

$L_k = \text{Lateness}$  pekerjaan ke- $k$

### Perhitungan Manual SOS

Pada awalnya dilakukan Perhitungan manual dengan contoh tabel variabel pada Tabel 1. Perhitungan manual diharapkan dapat memberikan

**Tabel 1.** Contoh tabel variabel algoritma SOS

Urutan Pengerjaan	Deskripsi Pekerjaan	Jumlah Batang	Treatment	Kebutuhan Keranjang	Jenis Warna	Jumlah Batang Setiap Hollow	Jenis Packing	Tanggal Terima
1	GDL	115	MF	4	9003	24	Profil p	01/27/20
2	TIO	40	MF	1	Silver White	3	Profil	02/04/20
3	Juanda	54	MF	1	7004	6	Plat p	02/10/20
4	Juanda	32	MF	1	7021	8	Plat p	02/03/20
5	Kulon Progo	94	MF	2	Silver Champ	5	Plat p	01/15/20

gambaran dari jalannya algoritma SOS. Perhitungan manual menggunakan contoh permasalahan dengan jumlah pekerjaan lima. Variabel dari algoritma SOS adalah karakteristik dari pesanan. Tabel 1 merupakan contoh variabel algoritma SOS. Tanggal penyelesaian setiap pekerjaan adalah tujuh hari kerja setelah tanggal terima. Kebijakan tanggal penyelesaian ini mengikuti kebijakan dari perusahaan sendiri.

Parameter yang digunakan dalam algoritma SOS adalah vektor solusi. Vektor solusi adalah kombinasi urutan pekerjaan sejumlah *eco size* yang ditetapkan sejak awal. *Eco size* dari perhitungan manual adalah 10. Setiap kombinasi pada vektor solusi adalah X individual dari algoritma. Iterasi pada perhitungan manual ini sebanyak satu iterasi. Perhitungan manual dari algoritma SOS sendiri melalui lima langkah perhitungan.

Pertama ialah mengisi data pekerjaan pada tabel variabel dan akan menghasilkan *lateness* dari setiap pekerjaan. *Lateness* pada permasalahan ini adalah selisih dari waktu pengerjaan dengan waktu yang tersedia untuk menyelesaikan pekerjaan. Waktu pengerjaan didapatkan dari hasil penjumlahan waktu proses produksi. Waktu yang tersedia untuk menyelesaikan pekerjaan sama dengan tanggal penyelesaian pekerjaan. Tabel 4 merupakan contoh dari penentuan *lateness* pekerjaan. Fungsi *max* pada Excel digunakan pada kolom *lateness* (hari) untuk mengetahui maksimum *lateness* dari urutan pekerjaan. Karakteristik pekerjaan menghasilkan waktu proses pekerjaan dari waktu proses *treatment*

hingga *packing* selesai. Tanggal awal dari urutan pengerjaan urutan pekerjaan yang ada pada Tabel 2 adalah 3 Februari 2020.

Pekerjaan GDL pada Tabel 2 memiliki urutan pekerjaan pertama, sehingga dikerjakan pada menit nol pada tanggal 3 Februari 2020. Waktu penyelesaian pekerjaan (*finish*) dihasilkan dengan menjumlahkan waktu awal pengerjaan (*start*) dan waktu *packing* selesai. Waktu proses pekerjaan GDL hingga *packing* selesai ialah 286,5 menit, sehingga pekerjaan GDL selesai tanggal 3 Februari 2020 menit 286,5. Pekerjaan GDL memiliki *due date* 5 Februari 2020, sehingga *lateness* dari pekerjaan GDL adalah minus dua. Waktu pengerjaan awal (*start*) pekerjaan selanjutnya akan dikerjakan pada saat pekerjaan GDL selesai.

Kedua ialah menghasilkan vektor solusi secara acak sesuai dengan jumlah pekerjaan dari Tabel 2. Proses menghasilkan vektor solusi ialah dengan *generate* data kontinu secara acak. Jumlah data kontinu untuk satu vektor X individual mengikuti jumlah pekerjaan yang ada. Data kontinu tersebut kemudian diubah menjadi diskrit dengan fungsi *rank.eq* pada Excel. Fungsi *rank.eq* memberikan peringkat dari nilai kontinu. Nilai data diskrit terkecil menyatakan urutan pekerjaan pertama dan nilai data diskrit terbesar menyatakan pekerjaan akhir.

Contoh vektor solusi dapat dilihat pada Tabel 3. Vektor  $X_i$  pada Tabel 3 terdiri dari 0,688691; 0,524352; 0,546517; 0,897632 dan 0,766307. Data kontinu ini kemudian diubah menjadi diskrit dengan

**Tabel 2.** Penentuan *lateness* pekerjaan

Urutan Pengerjaan	Deskripsi Pekerjaan	Waktu Treatment (menit)	Waktu Proses Powder Coating hingga Curing (menit)	Waktu Packing selesai (menit)	Due Date	Start (menit)	Tanggal Start	Finish (menit)	Tanggal Finish	Lateness (hari)
1	GDL	123	71,3	286,5	02/05/20	0,0	2/3/20	286,5	02/03/20	-2
2	TIO	94	86,3	124,5	02/13/20	286,5	2/3/20	366,0	02/03/20	-8
3	Juanda	94	68,0	146,0	02/19/20	366,0	2/3/20	27,5	02/04/20	-13
4	Juanda	94	59,7	107,5	02/12/20	27,5	2/4/20	124,5	02/04/20	-6
5	Kulon Progo	94	84,7	217,8	01/24/20	124,5	2/4/20	328,3	02/04/20	7

**Tabel 3.** Contoh vektor solusi perhitungan manual

Vektor	Komponen Vektor					<i>Lateness Max</i>
$X_1$	0,68 (3)	0,52 (5)	0,54 (4)	0,89 (1)	0,76 (2)	6
$X_2$	0,86 (2)	0,76 (3)	0,07 (5)	0,42 (4)	0,96 (1)	7
$X_3$	0,30 (5)	0,92 (2)	0,37 (4)	0,88 (3)	0,98 (1)	6
$X_4$	0,62 (4)	0,81 (2)	0,98 (1)	0,13 (5)	0,80 (3)	7
$X_5$	0,36 (4)	0,94 (1)	0,58 (2)	0,47 (3)	0,11 (5)	7
$X_6$	0,57 (4)	0,63 (1)	0,53 (5)	0,60 (2)	0,59 (3)	7
$X_7$	0,95 (1)	0,31 (4)	0,38 (3)	0,10 (5)	0,44 (2)	7
$X_8$	0,93 (1)	0,41 (3)	0,56 (2)	0,20 (4)	0,15 (5)	7
$X_9$	0,42 (4)	0,79 (3)	0,37 (5)	0,88 (1)	0,85 (2)	6
$X_{10}$	0,16 (4)	0,53 (2)	0,05 (5)	0,17 (3)	0,67 (1)	6

fungsi *rank.eq* pada Excel. Urutan pengerjaan dapat dilihat dari hasil diskritnya dari kiri ke kanan. Setiap X individual memiliki nilai maksimum *lateness*. Setelah dihasilkan vektor solusi, selanjutnya ialah menentukan X terbaik awalnya. Kombinasi yang menghasilkan *lateness* maksimum terkecil akan menjadi X terbaik. Apabila terdapat lebih dari satu nilai *lateness* maksimum terkecil yang sama, maka X terbaik dapat berdasarkan salah satunya. X terbaik vektor solusi di atas ialah  $X_1$  dengan nilai *lateness* maksimum enam.

Ketiga ialah menghasilkan vektor solusi mutualisme. Pada tahap mutualisme, terlebih dahulu ditentukan vektor  $X_i$  dan  $X_j$ . Nilai  $X_i \neq X_j$ , yang mana nilai  $i$  adalah nilai iterasi dan  $j$  nilai acak satu hingga *eco size*. Pada tahap mutualisme perhitungan manual,  $j$  terpilih adalah tiga.  $X_i$  dan  $X_j$  terpilih pada tahap mutualisme akan menghasilkan *mutual vector*. *Mutual vector* didapatkan dengan menggunakan Persamaan 3. Hasil *mutual vector* perhitungan manual dapat dilihat pada Tabel 4. Selanjutnya menentukan nilai  $X_i$  baru dan  $X_j$  baru.

**Tabel 4.** Hasil *mutual vector* perhitungan manual

Vektor	Komponen Vektor				
$X_1$	0,68	0,52	0,54	0,89	0,76
$X_3$	0,30	0,92	0,37	0,88	0,98
<i>Mutual Vector</i>	0,49	0,72	0,45	0,89	0,87

Nilai kontinu pada vektor diubah menjadi diskrit. Tujuannya ialah untuk mendapatkan *lateness* dari urutan kerja baru. Pemeriksaan *lateness* dibantu dengan mengisi tabel variabel sesuai urutan yang ingin diperiksa. Nilai *lateness* dari  $X_i$  baru dan  $X_j$  baru

dibandingkan dengan nilai *lateness* dari  $X_i$  lama dan  $X_j$  lama untuk menentukan mana yang akan menjadi vektor solusi yang baru. Nilai yang terkecil dari keduanya akan dipertahankan menjadi vektor solusi pada fase mutualisme. Apabila *lateness* X lama dan X baru bernilai sama, maka vektor solusi berdasarkan salah satunya.

Perhitungan vektor solusi fase mutualisme dapat dilihat pada Tabel 5. X mutualisme merupakan vektor solusi baru dari hasil perbandingan antara X lama dengan X baru. Nilai maksimum *lateness*  $X_i$  lama lebih kecil dari  $X_i$  baru, sehingga nilai vektor  $X_i$  lama dipertahankan. Nilai maksimum *lateness*  $X_j$  lama lebih kecil dari  $X_j$  baru, sehingga nilai vektor  $X_j$  lama dipertahankan.

**Tabel 5.** Perhitungan vektor solusi fase mutualisme

Vektor	Komponen Vektor					<i>Lateness Max</i>
	1	2	3	4	5	
$X_i$ baru	$X_1$ 0,41 (4)	0,34 (2)	0,48 (3)	0,72 (1)	0,44 (5)	7
$X_j$ baru	$X_3$ 0,27 (4)	-0,02 (5)	0,36 (3)	0,74 (1)	0,57 (2)	7
$X_i$ mutualisme	$X_1$ 0,68 (3)	0,52 (5)	0,54 (4)	0,89 (1)	0,76 (2)	6
$X_j$ mutualisme	$X_3$ 0,30 (5)	0,92 (2)	0,37 (4)	0,88 (3)	0,98 (1)	6

Vektor solusi yang diperbarui di tahap ini adalah vektor solusi dari hasil perbandingan dari X lama dengan X baru hasil mutualisme. Urutan pekerjaan yang baru pada vektor solusi yang baru akan berpotensi menghasilkan X terbaik yang baru. Vektor solusi mutualisme dari perhitungan manual dapat dilihat pada Tabel 6. X terbaik vektor solusi mutualisme perhitungan manual ialah  $X_1$  dengan nilai maksimum keterlambatan enam.

**Tabel 6.** Vektor solusi mutualisme

Vektor	Komponen Vektor					<i>Lateness Max</i>
$X_1$	0,68	0,52	0,54	0,89	0,76	6
$X_2$	0,86	0,76	0,07	0,42	0,96	6
$X_3$	0,30	0,92	0,37	0,88	0,98	7
$X_4$	0,62	0,81	0,98	0,13	0,80	7
$X_5$	0,36	0,94	0,58	0,47	0,11	7
$X_6$	0,57	0,63	0,53	0,60	0,59	7
$X_7$	0,95	0,31	0,38	0,10	0,44	7
$X_8$	0,93	0,41	0,56	0,20	0,15	7
$X_9$	0,42	0,79	0,37	0,88	0,85	6
$X_{10}$	0,16	0,53	0,05	0,17	0,67	6

Keempat ialah menghasilkan vektor solusi komensalisme. Pada tahap komensalisme, terlebih

dahulu ditentukan vektor  $X_i$  dan  $X_j$ . Pada tahap komensalisme perhitungan manual,  $j$  terpilih adalah enam. Vektor terpilih digunakan untuk menghasilkan  $X_i$  baru menggunakan Persamaan 4.

Nilai  $X_i$  baru dibandingkan dengan nilai  $X_i$  lama. Vektor dengan keterlambatan maksimum yang lebih kecil akan dipertahankan dan menjadi  $X$  komensalisme. Apabila *lateness*  $X$  lama dan baru bernilai sama, maka  $X$  komensalisme berdasarkan salah satunya. Hasil perbandingan vektor solusi fase komensalisme dapat dilihat pada Tabel 7.  $X$  komensalisme menunjukkan hasil perbandingan antara  $X_i$  baru dengan  $X_i$  lama. Nilai maksimum *lateness*  $X_i$  (enam hari) sama dengan  $X_i$  baru (enam hari), sehingga nilai vektor  $X_i$  lama dipertahankan.

**Tabel 7.** Perhitungan Vektor Solusi Fase Komensalisme

Vektor	Komponen Vektor	<i>Lateness</i>					
		1	2	3	4	5	<i>Max</i>
$X_i$ baru	$X_l$	0,78	0,27	0,49	0,76	0,53	6
		(1)	(5)	(4)	(2)	(3)	
$X$ komensalisme	$X_l$	0,68	0,52	0,54	0,89	0,76	6
		(3)	(5)	(4)	(1)	(2)	

Vektor solusi akan diperbarui mengikuti hasil  $X$  komensalisme. Langkah selanjutnya ialah mengubah nilai kontinu vektor menjadi diskrit. Tujuannya ialah mendapatkan *lateness* dari urutan kerja baru. Pengecekan *lateness* dibantu dengan mengisi tabel variabel sesuai urutan yang ingin diperiksa. Urutan pekerjaan yang baru akan berpotensi menghasilkan  $X$  terbaik yang baru.  $X$  terbaik dari vektor solusi komensalisme perhitungan manual ialah  $X_l$  dengan nilai maksimum keterlambatan enam.

Kelima ialah menghasilkan vektor solusi parasitisme. Pada tahap parasitisme, terlebih dahulu ditentukan vektor  $X_i$  dan  $X_j$  dengan Persamaan 5 dan 6. Pada tahap parasitisme perhitungan manual,  $j$  terpilih adalah dua. Vektor  $X_j$  terpilih dibandingkan dengan *parasite vector*. *Parasite vector* dihasilkan melalui mutasi  $X_i$ , dengan cara menghasilkan angka acak dari satu hingga jumlah pekerjaan (dalam contoh perhitungan manual ini adalah empat). Hasil acak tersebut menjadi komponen vektor solusi  $X_i$  yang akan diubah menjadi nilai kontinunya. Mengubah nilai kontinu komponen vektor terpilih dilakukan dengan generate angka acak mulai nol hingga satu (fungsi  $rand(0,1)$ ). *Parasite vector* berpotensi menghasilkan urutan pekerjaan dan maksimum keterlambatan baru.

Maksimum keterlambatan dari *parasite vector* dan vektor  $X_j$  kemudian dibandingkan. Vektor dengan

nilai maksimum keterlambatan yang lebih kecil akan menjadi  $X$  parasitisme. Hasil perbandingan vektor solusi parasitisme dapat dilihat pada Tabel 8.  $X$  parasitisme menunjukkan vektor terpilih hasil perbandingan antara  $X_j$  dengan *parasite vector*. Nilai maksimum *lateness*  $X_j$  sama dengan *parasite vector*, sehingga nilai vektor  $X_j$  dipertahankan.

**Tabel 8.** Perhitungan Vektor Solusi Fase Parasitisme

Vektor	Komponen Vektor	<i>Lateness</i>					
		1	2	3	4	5	<i>Max</i>
<i>parasite vector</i>	$X_l$	0,68	0,52	0,54	0,12	0,76	7
		(1)	(4)	(2)	(5)	(3)	
$X_j$ parasitisme	$X_2$	0,86	0,76	0,07	0,42	0,96	7
		(2)	(3)	(5)	(4)	(1)	

Vektor solusi akan diperbarui mengikuti hasil  $X$  parasitisme. Urutan pekerjaan yang baru akan berpotensi menghasilkan  $X$  terbaik yang baru.  $X$  terbaik dari vektor solusi parasitisme perhitungan manual ialah  $X_l$  dengan nilai maksimum keterlambatan enam. Setelah iterasi pertama menghasilkan nilai akhir, maka dilanjutkan dengan iterasi selanjutnya. Vektor solusi awal iterasi selanjutnya menggunakan hasil akhir iterasi sebelumnya. Simulasi akan berhenti apabila iterasi sudah mencapai jumlah iterasi yang sudah ditentukan sejak awal. Hasil *lateness* maksimum terkecil dari iterasi akhir merupakan usulan dari kombinasi terbaik. Urutan pekerjaan terbaik hasil perhitungan manual iterasi satu adalah 1, 5, 3, 2, 4 dengan maksimum keterlambatan enam.

Simulasi program algoritma dijalankan setelah perhitungan manual selesai dengan *eco size* 50 dan 20 replikasi. Penentuan jumlah *eco size* dan replikasi dilakukan dengan membandingkan hasil penjadwalan SOS satu replikasi dengan 20 replikasi. Tujuan dari perbandingan ini untuk melihat apakah jumlah replikasi dapat mempengaruhi hasil. Hasil perbandingan replikasi dapat dilihat pada Tabel 9.

**Tabel 9.** Hasil perbandingan replikasi

Jumlah Replikasi SOS	Jumlah <i>Generate Job</i>	Waktu Komputasi	<i>Lateness Max</i> (hari)
1 Replikasi	50	5 menit, 47 detik	10
	75	5 menit, 52 detik	13
	100	6 menit, 21 detik	13
20 Replikasi	50	136 menit, 29 detik	9
	75	139 menit, 10 detik	13
	100	159 menit, 12 detik	13

Hasil perbandingan replikasi SOS menunjukkan perbedaan hasil antara satu replikasi dengan 20 replikasi. Pada jumlah *generate job* 50 terbukti 20 replikasi menghasilkan *lateness* maksimum yang lebih optimal dibandingkan dengan satu replikasi.

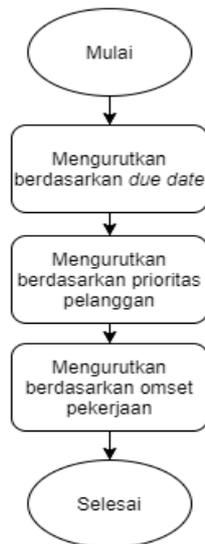
**Verifikasi Algoritma SOS**

Verifikasi dilakukan untuk mengetahui ketepatan program yang telah dibuat. Verifikasi algoritma dilakukan dengan membandingkan hasil program dengan perhitungan manual. Nilai acak dalam setiap iterasi ditentukan sejak awal.

Hasil perbandingan menunjukkan kesesuaian antara program dengan perhitungan manualnya. Kesesuaian ini menunjukkan bahwa program telah berjalan sesuai dengan algoritma SOS. Nilai kontinu vektor awal dari program dan perhitungan manual adalah sama.

**Algoritma EDD**

Usulan penjadwalan algoritma EDD adalah EDD yang dimodifikasi dengan mempertimbangkan kebijakan perusahaan. Parameter dari algoritma EDD adalah variabel *input* yang telah diisi oleh *user*. Penjadwalan EDD dilakukan dengan kriteria sortir sesuai kebijakan penjadwalan perusahaan. Alur penjadwalan EDD produksi perusahaan dapat dilihat pada Gambar 3.



**Gambar 3.** Alur penjadwalan EDD produksi perusahaan

Selanjutnya hasil replikasi SOS dengan data *generate job* 50, 75, dan 100 dibandingkan dengan hasil penjadwalan EDD. Perbandingan dilakukan untuk mengetahui apakah usulan penjadwalan SOS lebih baik dari penjadwalan EDD atau sebaliknya.

Perbandingan usulan penjadwalan dapat dilihat pada Tabel 10.

**Tabel 10.** Perbandingan usulan penjadwalan

Jumlah Replikasi SOS	Jumlah <i>Generate Job</i>	Waktu Komputasi	<i>Lateness Max</i> (hari)
1 Replikasi	50	5 menit, 47 detik	10
	75	5 menit, 52 detik	13
	100	6 menit, 21 detik	13
20 Replikasi	50	136 menit, 29 detik	9
	75	139 menit, 10 detik	13
	100	159 menit 12 detik	13
EDD	50	2 detik	6
	75	2 detik	5
	100	2 detik	8

Hasil perbandingan menunjukkan bahwa usulan penjadwalan dengan algoritma EDD lebih optimal bila dilihat dari hasil minimum maksimum *lateness*. Penjadwalan dengan algoritma EDD juga terbukti lebih efisien bila dilihat dari waktu komputasinya.

**Perbandingan Hasil Algoritma Penjadwalan**

Variabel dari simulasi program berdasarkan data pesanan perusahaan bulan Februari 2020. Maksimum keterlambatan dari hasil penjadwalan SOS adalah enam hari dengan waktu komputasi 141 menit, 30 detik. Maksimum keterlambatan dari hasil penjadwalan EDD adalah enam hari. Hasil keterlambatan maksimum algoritma SOS dan EDD bernilai sama, yakni enam hari. Waktu komputasi untuk program SOS lebih lama dari program EDD. Oleh karena itu, algoritma EDD merupakan usulan penjadwalan yang lebih efisien.

Usulan penjadwalan dengan algoritma EDD kemudian dibandingkan dengan kondisi aktual penjadwalan perusahaan pada bulan tersebut. Penjadwalan perusahaan aktual menghasilkan maksimum keterlambatan tujuh hari. Algoritma EDD menghasilkan maksimum keterlambatan lebih kecil dari penjadwalan perusahaan. Disimpulkan algoritma EDD menjadi usulan terbaik dan layak untuk diimplementasikan. Hasil dari penjadwalan SOS tidak lebih baik dari EDD karena rancangan penjadwalan SOS belum memperhatikan kebijakan dari penjadwalan perusahaan. Kondisi ini menyebabkan vektor solusi yang diberikan terjebak dalam optimum lokal. Data kontinu SOS yang dibawa pada data diskrit urutan *job generate random* dalam algoritma SOS tidak banyak berubah atau mendekati kebutuhan perusahaan.

## Simpulan

Tujuan penelitian ini adalah memberikan usulan penjadwalan produksi dengan membandingkan dua algoritma penjadwalan untuk meminimalkan maksimum *lateness*. Algoritma yang diuji adalah algoritma SOS dan EDD. *Performance measure* dalam membandingkan kedua algoritma adalah maksimum keterlambatan. Variabel algoritma penjadwalan menggunakan karakteristik dari pesanan yang ada.

Tahap awal dalam menghasilkan usulan penjadwalan ialah membuat perhitungan manual algoritma. Perhitungan manual dilakukan untuk mengetahui gambaran dari jalannya algoritma SOS. *Eco size* dalam perhitungan manual SOS adalah 10 dengan jumlah *job* lima dan iterasi sebanyak satu. Jumlah *generate* pekerjaan pada simulasi program sebanyak 50, 75, dan 100. *Eco size* dari simulasi program SOS adalah 50 dengan jumlah replikasi sebanyak satu dan 20. Hasil simulasi program SOS untuk jumlah pekerjaan 50, 75, dan 100 adalah maksimum keterlambatan sembilan hari, 13 hari, dan 13 hari.

Data pekerjaan yang sama dijadwalkan dengan algoritma EDD. Hasil dari penjadwalan menggunakan algoritma EDD untuk jumlah pekerjaan 50, 75, dan 100 adalah maksimum keterlambatan enam hari, lima hari, dan delapan hari. Waktu komputasi yang dibutuhkan dalam penjadwalan algoritma EDD adalah dua detik. Hasil kedua algoritma penjadwalan dibandingkan dengan data pesanan perusahaan periode Februari 2020.

Kedua algoritma memberikan hasil maksimum keterlambatan enam hari. Namun waktu komputasi penjadwalan EDD lebih efisien dari penjadwalan SOS. Algoritma EDD disimpulkan menjadi usulan penjadwalan terbaik.

Algoritma penjadwalan SOS dan EDD memberikan hasil yang sama optimalnya. Namun waktu komputasi penjadwalan EDD lebih efisien dari penjadwalan SOS. Algoritma EDD disimpulkan menjadi usulan penjadwalan terbaik. Algoritma usulan terbaik dibandingkan dengan kondisi aktual penjadwalan perusahaan. Hasil penjadwalan EDD lebih optimal dari penjadwalan perusahaan aktual. Penjadwalan EDD layak untuk diimplementasikan pada penjadwalan perusahaan.

## Saran

Saran bagi peneliti selanjutnya adalah memberikan prioritas pada proses awal *generate random* dari vektor solusi awal algoritma SOS. Prioritas yang diberikan perlu mengikuti kebijakan yang dimiliki perusahaan. Diharapkan saran ini dapat memberi hasil penjadwalan SOS yang lebih optimum dan efisien dalam memenuhi kebutuhan perusahaan.

## Daftar Pustaka

1. Cheng, M. Y., and Prayogo, D., Symbiotic Organisms Search: a New Metaheuristic Optimization Algorithm, *Computers & Structures*, 139, 2014, pp. 98-112.
2. Bedworth, D., and Bailey, J., *Integrated Production Control Systems*, John Wiley and Sons Inc., New York, 1987.