

# Penyelesaian Masalah Penempatan Fasilitas dengan Algoritma Estimasi Distribusi dan Particle Swarm Optimization

Amalia Utamima<sup>1)</sup>, Angelia Melani Adrian<sup>2)</sup>

<sup>1)</sup>*Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember  
Kampus ITS Sukolilo, Surabaya, 60111*

<sup>1)</sup>amalia@its-sby.edu

<sup>2)</sup>*Department of Industrial Management, National Taiwan University of Science and Technology  
Taipei City 106, Taiwan, ROC*

<sup>2)</sup>melaning21@yahoo.com

---

**Abstrak**—Masalah penempatan fasilitas pada garis lurus dikenal sebagai problem Penempatan Fasilitas pada Satu Baris (PFSB). Tujuan PFSB, yang dikategorikan sebagai masalah NP-Complete, adalah untuk mengatur tata letak sehingga jumlah jarak antara pasangan semua fasilitas bisa diminimalisir. Algoritma Estimasi Distribusi (EDA) meningkatkan kualitas solusi secara efisien dalam beberapa pengoperasian pertama, namun keragaman dalam solusi hilang secara pesat ketika semakin banyak iterasi dijalankan. Untuk menjaga keragaman, hibridisasi dengan algoritma meta-heuristik diperlukan. Penelitian ini mengusulkan EDAPSO, algoritma yang terdiri dari hibridisasi EDA dan Particle Swarm Optimization (PSO). Tujuan dari penelitian ini yaitu untuk menguji performa algoritma EDAPSO dalam menyelesaikan PFSB. Kinerja EDAPSO yang diuji dalam 10 masalah benchmark PFSB dan EDAPSO berhasil mencapai solusi optimal.

**Kata kunci**—penempatan fasilitas, algoritma estimasi distribusi, particle swarm optimization

---

**Abstract**—The layout positioning problem of facilities on a straight line is known as Single Row Facility Layout Problem (PFSB). Categorized as NP-Complete problem, PFSB aim to arrange the layout so that the sum of distances between all facilities' pairs can be minimized. Estimation of Distribution Algorithm (EDA) improves the solution quality efficiently in first few runs, but the diversity lost grows rapidly as more iterations are run. To maintain the diversity, hybridization with meta-heuristic algorithms is needed. This research proposes EDAPSO, an algorithm which consists of hybridization of EDA and Particle Swarm Optimization (PSO). The objective of this research is to test the performance of EDAPSO algorithm for solving PFSB. EDAPSO's performance is tested in 10 benchmark problems of PFSB and it successfully achieves optimum solution.

**Keywords**— facility layout, estimation distribution algorithm, particle swarm optimization

**Article history:**

Received 16 January 2016; Received in revised form 10 March 2016; Accepted 11 March 2016; Available online 30 April 2016

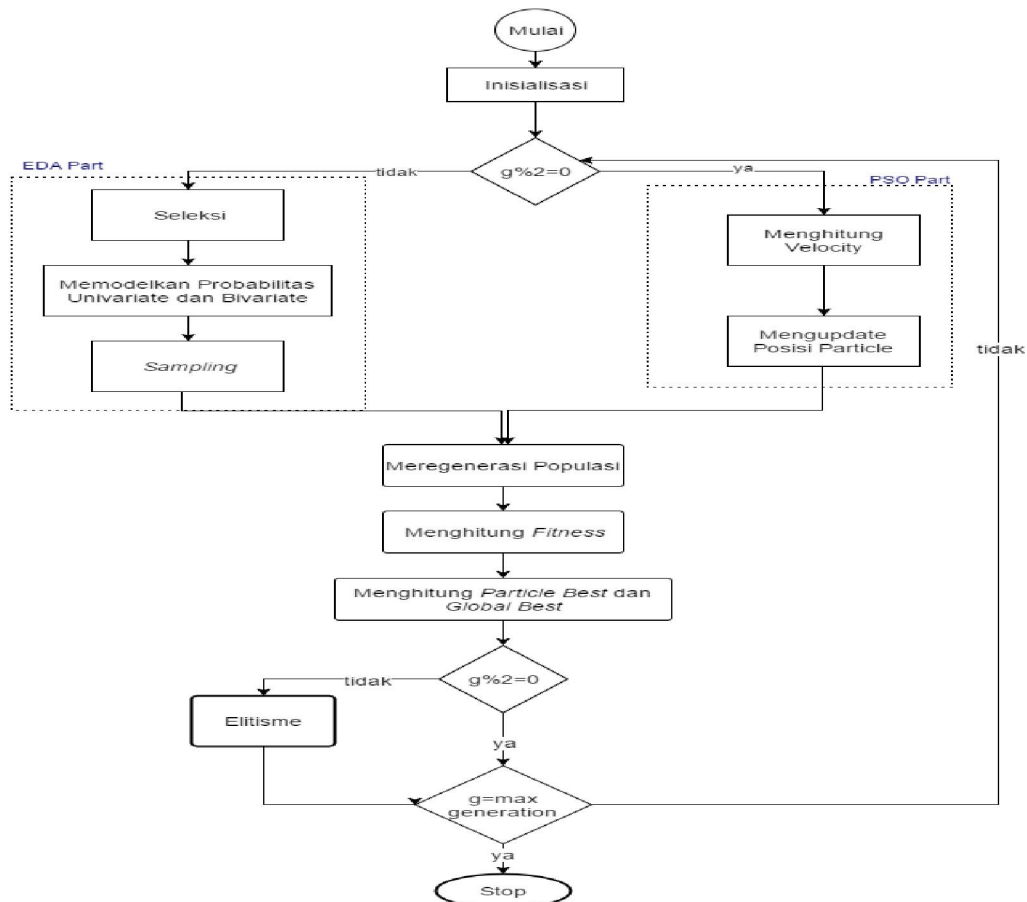
---

## I. PENDAHULUAN

Masalah Penempatan Fasilitas, dikategorikan sebagai *Quadratic Assignment Problem*, memperhatikan penataan sejumlah fasilitas di ruang tertentu untuk memenuhi fungsi tujuan. Penempatan fasilitas yang dirancang secara efektif dapat mengurangi waktu tunggu di pabrik, dan meningkatkan jumlah produk yang sukses dikirim. Dengan demikian, meningkatkan produktivitas secara keseluruhan dan efisiensi pabrik.

Permasalahan Penempatan Fasilitas pada Satu Baris atau *Single Row Facility Layout Problem*(PFSB) terjadi ketika beberapa produk

dengan volume produksi yang berbeda dan rute proses yang berbeda perlu diproduksi. Tujuannya adalah untuk mengatur fasilitas sehingga dapat meminimalkan jumlah jarak antara semua pasangan fasilitas (Amaral, 2006). Karena PFSB terbukti menjadi masalah NP-Complete, metode yang tepat diterapkan untuk kasus besar memakan banyak waktu. Maka, metode heuristik telah dibangun untuk memperoleh solusi optimal untuk masalah ini (Samarghandi & Eshghi, An efficient tabu algorithm for the single row facility layout problem, 2010).



Gambar 1. Kerangka kerja EDAPSO

Penelitian baru-baru ini telah mencoba untuk memecahkan PFSB dengan pendekatan meta-heuristik. Beberapa diantaranya, Samarghandi et al. menggunakan *Particle Swarm Optimization (PSO)* dan *Tabu Search*, untuk mencari tahu solusi efisien untuk PFSB (Samarghandi, Taabayan, & Jahantigh, A particle swarm optimization for the single row facility layout problem, 2010) (Samarghandi & Eshghi, An efficient tabu algorithm for the single row facility layout problem, 2010). (Datta, Amaral, & Figueira, 2011) menawarkan permutasi berbasis algoritma genetika untuk mendapatkan solusi optimal untuk PFSB. Dengan memecahkan PFSB secara efektif, suatu algoritma diharapkan juga bisa berhasil untuk memecahkan kasus yang berbeda pada problem penempatan fasilitas.

Algoritma Estimasi Distribusi atau *Estimation Distribution Algorithm (EDA)* adalah teknik optimasi stokastik yang mengeksplorasi ruang solusi potensial dengan memanfaatkan ketergantungan antar variabel dan melakukan proses sampel model probabilistik pada calon solusi yang menjanjikan (Hauschild & Pelikan, 2011). Oleh karena itu, EDA bisa memecahkan masalah optimasi kompleks secara efisien (Zhang & Muehlenbein, 2004).

EDA meningkatkan kualitas solusi secara efisien dalam beberapa pengoperasian pertama, namun keragaman dalam solusi hilang secara pesat ketika semakin banyak iterasi dijalankan. Untuk menjaga keragaman, hibridisasi dengan algoritma meta-heuristik diperlukan. EDA akan digunakan untuk mengkarakterisasi *parent solution* dan kemudian mencari di sekitar ruang solusi saat ini. Setelah itu, meta-heuristik (dalam penelitian ini adalah PSO) memperkenalkan solusi baru ke dalam populasi untuk mempertahankan keragaman, yang dapat menghindari konvergensi prematur EDA (Chen, Chen, Chang, & Chen, 2012). EDA dan PSO dikombinasikan dan untuk membentuk suatu algoritma baru yang disebut EDAPSO. Tujuan dari penelitian ini yaitu untuk menguji performa algoritma EDAPSO dalam menyelesaikan PFSB.

## II. METODOLOGI

Secara umum, penelitian ini melakukan dua tahapan yaitu membangun algoritma EDAPSO dan mengaplikasikan EDAPSO untuk menyelesaikan PFSB.

Penelitian ini menganggap PFSB dengan ukuran fasilitas yang berbeda. Diketahui panjang  $l_i$  dari setiap fasilitas  $i$  dan  $n \times n$  matrix  $C = [c_{ij}]$ , dimana  $c_{ij}$  adalah beban lalu lintas antara fasilitas  $i$  dan  $j$ . Jarak antara dua fasilitas

diambil antara titik pertengahan. ABSMODEL (Heragu & Kusiak, 1991) adalah model umum untuk memecahkan PFSB. ABSMODEL diilustrasikan dalam persamaan (1).

$$\min z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} d_{ij} \quad (1)$$

$$\begin{aligned} \text{s.t.} : d_{ij} &\geq \frac{1}{2}(l_i + l_j) + s_{ij}; \\ i &= 1, 2, \dots, n-1; j = i+1, \dots, n \\ d_{ij} &\geq 0; \\ i &= 1, 2, \dots, n-1; j = i+1, \dots, n \end{aligned}$$

Dalam ABSMODEL,  $d_{ij}$  merupakan jarak antara pusat-pusat fasilitas  $i$  dan  $j$ , sedangkan  $s_{ij}$  adalah izin yang diperlukan atau kesenjangan antara dua fasilitas. Perhitungan  $d_{ij}$  ditunjukkan dalam persamaan (2).

$$d_{ij} = \frac{l_i + l_j}{2} + D_{ij} \quad (2)$$

$D_{ij}$  adalah ruang antara fasilitas  $i$  dan  $j$ .  $D_{ij}$  belum tentu sama dengan  $s_{ij}$ . Jika fasilitas  $k$  ditempatkan di antara fasilitas  $i$  dan  $j$  dengan  $s_{ij} = 0$  maka  $D_{ij} = l_k$ . Dalam penelitian ini, diasumsikan nilai  $s_{ij}$  adalah 0.

Kerangka kerja EDAPSO disediakan pada Gambar 1. Algoritma ini menjalankan prosedur EDA dan PSO secara bergantian. Algoritma dimulai dengan inialisasi dari semua variabel. Prosedur EDA kemudian mulai berjalan ketika generasi saat ini dapat dibagi dengan 2. Jika tidak, prosedur PSO dilakukan. Prosedur PSO terdiri dari perhitungan kecepatan dan pembaruan posisi partikel. Setelah perhitungan *fitness*, partikel yang terbaik dan nilai-nilai global terbaik dihitung.

Masih pada Gambar 1, Regenerasi populasi yang mengganti *parent* partikel dengan *children* partikel dilakukan setelah metode EDA atau PSO dijalankan. Selanjutnya, dilakukan perhitungan nilai *fitness* terhadap populasi baru ini yang diikuti oleh perhitungan *particle best* dan *global best*. Strategi elitisme dilakukan jika pada generasi berikutnya akan dijalankan prosedur EDA. Di akhir Gambar 1, terdapat pengecekan apakah generasi saat ini ( $g$ ) telah mencapai maksimum generasi (ditentukan sesuai dengan besarnya problem). Apabila  $g$  telah mencapai maksimum jumlah generasi maka algoritma akan berhenti, apabila belum, maka perulangan akan dilanjutkan.

#### A. Tahap Estimation Distribution Algorithm

Dalam permasalahan yang berisi  $n$  fasilitas yang diberi label dengan  $F^1, F^2, \dots, F^n$ , urutan  $X$  berisi nomor permutasi dari semua label.  $X$  dapat direpresentasikan dengan menetapkan fasilitas  $X[i]$  ke lokasi  $i$ , dimana  $i \in [1, n]$ .

Urutan yang berbeda pada  $X$  merupakan solusi tata letak yang berbeda.

Representasi kromosom dapat dilihat pada Tabel 1. Jumlah lokasi kromosom yang disesuaikan dengan jumlah fasilitas dalam masalah yang menjadi *benchmark*. Tabel 1 menunjukkan contoh representasi kromosom dengan 8 fasilitas.  $X[2] = 3$  berarti fasilitas 3 ditempatkan ke lokasi 2.

TABEL 1. ILUSTRASI KROMOSOM

i	1	2	3	4	5	6	7	8
X[i]	5	3	4	1	8	2	7	6

Dalam setiap generasi  $g$ , kelompok kromosom  $C$  yang memiliki *fitness* yang lebih baik akan dipilih. Kromosom diberi label  $X^1, X^2, \dots, X^C$ , di mana  $C$  adalah setengah dari jumlah populasi. Distribusi kromosom *parent* diartikan dengan *sampling* solusi baru dari model probabilistik univariat dan bivariat (Chen, Chen, Chang, & Chen, 2012). Solusi yang dihasilkan oleh dua model probabilistik dikenal sebagai *extended artificial particles*. Dalam penelitian ini diadopsi model probabilistik yang sama seperti yang dijelaskan (Chen, Chen, Chang, & Chen, 2012). Model univariat atau probabilistik ordinal,  $\phi_{i[i]}(g)$  dalam persamaan (3) menunjukkan pentingnya fasilitas dalam urutan. Hal ini menunjukkan berapa kali fasilitas  $i$  ditempatkan sebelum atau pada posisi  $[i]$  di generasi  $g$  saat ini.  $A_{i[i]}^k$  akan diset ke 1 jika fasilitas  $i$  ditempatkan sebelum atau pada posisi  $[i]$ . Selain itu, akan diatur ke 0.

$$\phi_{i[i]}(g) = \sum_{k=1}^C A_{i[i]}^k, i = 1, \dots, n; k = 1, \dots, C \quad (3)$$

Bivariat atau model probabilistik dependen  $\psi_{ij}$  dalam (4) merupakan berapa kali fasilitas  $j$  ditempatkan segera setelah fasilitas  $i$ .  $B_{ij}^k$  akan diset ke 1 jika fasilitas  $j$  ditempatkan di samping fasilitas  $i$ , jika tidak maka akan diatur ke 0. Nilai 0 dalam  $\phi_{i[i]}(g)$  dan  $\psi_{ij}(g)$  diganti dengan  $1/C$  untuk menjaga keragaman dari EDAPSO.

$$\psi_{ij}(g) = \sum_{k=1}^C B_{ij}^k, i, j = 1, \dots, n; k = 1, \dots, C \quad (4)$$

$P_{i[i]}(g)$  merupakan nilai probabilitas fasilitas  $i$  pada posisi  $[i]$ . Pemilihan fasilitas  $i$  yang memiliki nilai probabilitas lebih baik daripada fasilitas lain ketika informasi statistik kedua model probabilistik digunakan, menurut

$\phi_{i[i]}(g)$  dikalikan dengan  $\psi_{ij}(g)$ . Cara ini akan merangkum nilai-nilai probabilitas dari semua fasilitas yang belum ditetapkan yang bisa diatur pada posisi  $[i]$  (Chen, Chen, Chang, & Chen, 2012).

Untuk setiap *offspring*  $O^1, O^2, \dots, \text{ and } O^{2^C}$ , beberapa metode digunakan untuk menetapkan fasilitas ke lokasi yang ditentukan. Untuk memilih fasilitas di lokasi pertama, metode yang diusulkan secara acak memilih fasilitas pertama yang muncul dalam kromosom  $C$ , yaitu  $X$ . Misalnya untuk menetapkan fasilitas pertama ke  $O^1$ , jika  $X^2$  diambil secara acak, maka fasilitas pertama adalah dari  $X^3$ . Pendekatan ini lebih menjanjikan daripada yang diusulkan dalam (Chen, Chen, Chang, & Chen, 2012), yang memilih fasilitas acak. Untuk menempatkan fasilitas yang tersisa, diadopsi formula dalam (Chen, Chen, Chang, & Chen, 2012) yang ditunjukkan pada persamaan (5).

$$P_{i[i]}(g) = \frac{\phi_{i[i]}(g) \times \psi_{ij}(g)}{\sum_{f \in \Omega} (\phi_{i[i]}(g) \times \psi_{ij}(g))} \quad (5)$$

Dimana:

$[i] = 2, 3, \dots, n; i = 1, 2, \dots, n;$

$\Omega$ : himpunan fasilitas yang belum ditentukan.

Sebuah metode seleksi untuk memilih fasilitas dari  $\Omega$  dan meletakkannya di posisi  $[i]$  dimodifikasi dari (Chen, Chen, Chang, & Chen, 2012). Metode seleksi ini dilakukan dalam memproduksi setiap *offspring*. *Pseudocode* berikut menunjukkan prosedur penetapan.

1:	$F \leftarrow \emptyset$
2:	<b>for</b> $k=2$ <b>until</b> $n$ <b>do</b>
3:	$\theta \leftarrow U(0,1)$
4:	Select a facility $i$ satisfy $\theta \leq P_{i[i]}$ , where $i \in \Omega$
5:	$F(k) \leftarrow i$
6:	$\Omega \leftarrow \Omega \setminus i$
7:	<b>end for</b>

Gambar 2. Prosedur penetapan

Dimana:

$\Omega$  : Himpunan fasilitas yang ditetapkan.

$F$  : Himpunan fasilitas yang ditetapkan.  $F$  kosong di awal.

$\theta$  : Sebuah probabilitas acak diambil dari  $U(0,1)$

$i$  : Sebuah fasilitas yang dipilih oleh seleksi proporsional

$k$  : Indeks unsur posisi fasilitas yang

$n$  : Jumlah fasilitas.

### B. Tahap Particle Swarm Optimization

Prosedur PSO dimulai ketika generasi saat ini bernilai *n* dan dimulai dengan inisialisasi

dari semua parameter yang terkait. Pengaturan berat inersia yang dirujuk dari (Haupt & Haupt, 2004) tercantum pada persamaan (6)

$$\text{Inertia Weight} = \frac{\text{maxgen} - \text{currentgen}}{\text{maxgen}} \quad (6)$$

Selanjutnya, perhitungan kecepatan dari setiap partikel dioperasikan. Dalam standar PSO, kecepatan ditambahkan ke partikel pada setiap dimensi untuk memperbarui partikel, sehingga dalam hal ini digunakan ukuran jarak. Jika kecepatan lebih besar, partikel dapat menjelajahi jangkauan yang lebih jauh. Demikian pula, kecepatan baru dalam skenario permutasi merupakan kemungkinan bahwa perubahan partikel. Jika kecepatan lebih besar, partikel lebih mungkin untuk mengubah urutan permutasi baru. Rumus pembaruan kecepatan dalam persamaan (7) tetap sama, sedangkan proses update partikel berubah.

$$\begin{aligned} \vec{v}_i(t+1) &= w \times \vec{v}_i(t) + r_1 c_1 (\vec{x}_{pBest} - \\ \vec{x}_i(t)) &+ r_2 c_2 (\vec{x}_{gBest} - \vec{x}_i(t)) \end{aligned} \quad (7)$$

EDAPSO yang diajukan dalam makalah ini menggunakan teknik dalam memperbarui partikel berbasis permutasi berdasarkan konsep dari (Hu, Eberhart, & Shi, 2003).

## III. HASIL DAN PEMBAHASAN

Terdapat dua skenario yang dilakukan dalam percobaan untuk menguji metode yang diusulkan. Pertama, perbandingan metode yang diusulkan dengan EDA dasar dan PSO untuk memeriksa ketahanan ketika menyelesaikan PFSB. Dalam uji coba ini, digunakan sepuluh masalah *benchmark* yaitu P4, LW5, N6, S8, S8H, S9, S9H, S10, S11 dan LW11 yang dijelaskan dalam (Amaral, 2006). Jumlah fasilitas di setiap masalah diberi label sebagai  $n$ .

Dalam skenario pertama, dilakukan 10 uji coba untuk setiap algoritma dan mempertimbangkan hasil minimum dan tingkat kesalahan masing-masing algoritma dalam mencapai solusi optimal. Persentase tingkat kesalahan (*error*) yang diperoleh yaitu dengan membandingkan nilai rata-rata di 10 kali program berjalan dan nilai minimum yang dicapai dari ketiga algoritma. Selanjutnya, dalam skenario kedua, dibandingkan nilai minimum yang dicapai atau yang bisa disebut *Objective Function Value* (OFV). Hasil dari skenario pertama, kedua, dan ketiga masing-masing diberikan dalam Tabel 2 dan Tabel 3.

TABEL 2. PERBANDINGAN EDA, PSO, DAN EDAPSO

Problem	n	EDA		PSO		EDAPSO	
		min	error (%)	min	error (%)	min	error (%)
P4	4	<b>638.0</b>	<b>0.00%</b>	<b>638.0</b>	<b>0.00%</b>	<b>638.0</b>	<b>0.00%</b>
LW5	5	<b>151.0</b>	0.73%	<b>151.0</b>	0.23%	<b>151.0</b>	<b>0.00%</b>
S8	8	<b>801.0</b>	1.95%	<b>801.0</b>	2.76%	<b>801.0</b>	<b>0.42%</b>
S8H	8	<b>2324.5</b>	0.86%	<b>2324.5</b>	2.25%	<b>2324.5</b>	<b>0.47%</b>
S9	9	<b>2469.5</b>	2.23%	<b>2469.5</b>	<b>0.74%</b>	<b>2469.5</b>	2.21%
S9H	9	<b>4695.5</b>	0.47%	<b>4695.5</b>	5.74%	<b>4695.5</b>	<b>0.23%</b>
S10	10	<b>2781.5</b>	2.88%	<b>2781.50</b>	5.24%	<b>2781.5</b>	<b>2.79%</b>
S11	11	<b>6933.5</b>	4.93%	<b>6933.50</b>	4.04%	<b>6933.5</b>	<b>3.08%</b>
LW11	11	<b>6933.5</b>	4.71%	6973.50	3.91%	<b>6933.5</b>	<b>3.61%</b>
P15	15	6534.0	6.42%	6576.50	10.21%	<b>6305.0</b>	<b>4.82%</b>

TABEL 3. PERBANDINGAN DENGAN PENELITIAN SEBELUMNYA

Problem	n	OFV	OFV	OFV	OFV	Gap
		(Amaral, 2006)	(Samarghandi, Taabayan, & Jahantigh, 2010)	(Datta, Amaral, & Figueira, 2011)	EDAPSO	
P4	4	<b>638.0</b>	<b>638.0</b>	-	<b>638.0</b>	0.00%
LW5	5	<b>151.0</b>	<b>151.0</b>	<b>151.0</b>	<b>151.0</b>	0.00%
S8	8	<b>801.0</b>	<b>801.0</b>	-	<b>801.0</b>	0.00%
S8H	8	<b>2324.5</b>	<b>2324.5</b>	<b>2324.5</b>	<b>2324.5</b>	0.00%
S9	9	<b>2469.5</b>	<b>2469.5</b>	-	<b>2469.5</b>	0.00%
S9H	9	<b>4695.5</b>	<b>4695.5</b>	-	<b>4695.5</b>	0.00%
S10	10	<b>2781.5</b>	<b>2781.5</b>	<b>2781.5</b>	<b>2781.5</b>	0.00%
S11	11	<b>6933.5</b>	<b>6933.5</b>	<b>6933.5</b>	<b>6933.5</b>	0.00%
LW11	11	<b>6933.5</b>	<b>6933.5</b>	<b>6933.5</b>	<b>6933.5</b>	0.00%
P15	15	<b>6305.0</b>	<b>6305.0</b>	-	<b>6305.0</b>	0.00%

Dalam uji coba ini, digunakan pengaturan parameter yang sama untuk EDAPSO, EDA, dan PSO. Ukuran populasi dan generasi bervariasi tergantung pada jumlah fasilitas. Semakin tinggi jumlah fasilitas, semakin banyak populasi dan generasi yang diperlukan untuk mendapatkan solusi yang optimal. Selain itu, ditetapkan ukuran populasi sebagai  $4n$  dalam setiap masalah. Sedangkan jumlah maksimum generasi adalah  $10n$ . Tingkat elitisme adalah 10 persen untuk setiap generasi EDAPSO.

Dibandingkan dengan penelitian yang lain, nilai minimum atau OFV dicapai oleh EDAPSO menunjukkan kinerja yang sama dengan OFV penelitian sebelumnya dalam semua masalah *benchmark*. Hasil uji coba ini mencapai selisih 0.00% jika dibandingkan dengan OFV penelitian sebelumnya.

Tabel 2 menunjukkan perbandingan properti statistik antara metode yang diusulkan, EDA, dan PSO. Data yang dicetak tebal menunjukkan nilai-nilai minimum yang

ditemukan di antara semua algoritma. Dapat diamati pada Tabel 2 bahwa EDAPSO mempunyai nilai minimum lebih unggul untuk masalah LW11 sampai P15 dibandingkan dengan PSO, sedangkan untuk masalah P4 sampai S10 memiliki nilai minimal sama. Dibandingkan dengan EDA, EDAPSO memiliki nilai minimum yang lebih rendah pada P15. Selain itu, kita dapat melihat bahwa tingkat error dari EDAPSO sebagian besar memiliki nilai terendah pada problem benchmark dibandingkan dengan EDA dan PSO. Pada S9, nilai error yang dimiliki PSO lebih rendah daripada EDA dan EDAPSO, hal ini dimungkinkan terjadi mengingat bahwa metode PSO memiliki area pencarian yang luas dan terdapat probabilitas bahwa solusi yang didapat adalah nilai optimal dari suatu permasalahan.

Tabel 3 menyajikan nilai minimisasi yang disebut sebagai OFV dalam makalah sebelumnya, dibandingkan dengan EDAPSO. Nilai-nilai ini sangat penting karena PFSB bertujuan untuk meminimalkan fungsi tujuan

tersebut. OFV dari penelitian sebelumnya diperoleh dari (Amaral, 2006), (Samarghandi, Taabayan, & Jahantigh, A particle swarm optimization for the single row facility layout problem, 2010), dan (Datta, Amaral, & Figueira, 2011). Tanda *dash* ('-') pada Tabel 3 berarti bahwa hasil tidak tersedia dalam referensi yang dibandingkan. Dari Tabel 3 dapat terlihat bahwa OFV yang dihasilkan oleh EDAPSO dapat menyamai OFV yang dihasilkan oleh penelitian-penelitian sebelumnya.

#### IV. KESIMPULAN

Makalah ini mengusulkan *Estimation Distribution Algorithm Particle Swarm Optimization* (EDAPSO) untuk mencari solusi optimal untuk memecahkan masalah NP-Complete yaitu Fasilitas Row Tunggal Tata Letak atau *Single Row Facility Layout Problem* (PFSB). Hasil komputasi menunjukkan bahwa EDAPSO memberikan kinerja lebih baik dalam mendapatkan nilai-nilai minimum dan kesalahan minimal dalam sepuluh masalah *benchmark* yang digunakan.

Tidak seperti EDA yang menggunakan satu model probabilistik, EDAPSO menggunakan dua. EDA menggunakan model probabilistik univariat, sedangkan EDAPSO berlaku model probabilistik univariat dan bivariat. Penggunaan interaksi variabel dalam model probabilistik bivariat dapat mewakili informasi individual yang lebih baik untuk bagian EDA di EDAPSO. Hal ini memungkinkan EDAPSO untuk melakukan perhitungan lebih baik daripada EDA.

Partikel buatan yang dihasilkan dari informasi statistik global menunjukkan distribusi solusi yang baik dalam ruang pencarian. Oleh karena itu, kombinasi dari partikel buatan dengan operator PSO dapat meningkatkan kualitas solusi. Hal ini juga membuat EDAPSO tampil lebih baik daripada PSO standar.

Selain itu, jika dibandingkan dengan algoritma sebelumnya diterbitkan dalam literatur, EDAPSO mencapai kinerja yang sama di semua 10 masalah *benchmark*. Gap sebesar

0% menunjukkan bahwa EDAPSO selalu berhasil mencapai nilai minimum data acuan yang diberikan. Dengan demikian, dapat disimpulkan bahwa EDAPSO berhasil memecahkan PFSB untuk mencapai biaya minimum dengan kinerja yang baik.

#### DAFTAR PUSTAKA

- Amaral, A. R. (2006). On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173 (2), 508–518.
- Chen, Y. M., Chen, M. C., Chang, P. C., & Chen, S. H. (2012). Extended artificial chromosomes genetic algorithm for permutation flowshop scheduling problems. *Computers & Industrial Engineering*, 62 (2), 536-545.
- Datta, D., Amaral, A. R., & Figueira, J. (2011). Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213 (2), 388-394.
- Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms*. Ney Jersey: John Wiley & Sons.
- Hauschild, M., & Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1 (3), 111-128.
- Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problems. *European Journal of Operational Research*, 53 (1), 1–13.
- Hu, X., Eberhart, R. C., & Shi, Y. (2003). Swarm intelligence for permutation optimization: a case study of n-queens problem. *IEEE swarm intelligence symposium* (hal. 243–246). Indianapolis: IEEE.
- Samarghandi, H., & Eshghi, K. (2010). An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*, 205 (1), 98-105.
- Samarghandi, H., Taabayan, P., & Jahantigh. (2010). A particle swarm optimization for the single row facility layout problem. *Computers and Industrial Engineering*, 58 (4), 529-534.
- Zhang, Q., & Muehlenbein, H. (2004). On the Convergence of a Class of Estimation of Distribution Algorithms. *IEEE Trans. on Evolutionary Computation*, 8 (2), 127-136.