

**Manual Penggunaan Algoritma Evolusi  
Diferensial untuk Mengoptimasikan Tata Letak  
Fasilitas**

**Komarudin**



**Laboratorium Rekayasa, Simulasi dan Pemodelan Sistem  
Departemen Teknik Industri  
Fakultas Teknik Universitas Indonesia  
Kampus UI Depok  
Depok 16424  
Indonesia**

## 1. Daftar Isi

1. Daftar Isi .....	1
2. Pendahuluan .....	2
3. Dasar Teori.....	2
3.1 Teori Dasar Permasalahan Tata Letak Fasilitas dengan Luas Tak Sama (UA- FLP) .....	2
4. Tujuan Instruksional .....	5
5. Deskripsi Algoritma .....	5
6. Tahapan Penggunaan Algoritma .....	7
6.1 Membuka (Program) Algoritma .....	7
6.2 Menginput dan Meng-edit Data.....	7
6.3 Memulai Iterasi.....	9
6.4 Menghentikan Sementara dan Menghentikan Permanen .....	10
6.5 Menyimpan Hasil.....	11

## 2. Pendahuluan

Manual ini terbagi menjadi beberapa bagian yang perlu untuk dijelaskan sehingga memudahkan bagi pengguna (*user*) untuk menuju ke bagian yang dibutuhkan dalam menggunakan algoritma ini. Bagian-bagian dalam manual ini adalah bagian Dasar Teori, bagian Tujuan Instruksional, bagian Deskripsi Algoritma, dan bagian Tahapan Penggunaan Algoritma.

Bagian Dasar Teori menjelaskan tentang permasalahan yang dapat diselesaikan dengan algoritma ini. Istilah untuk permasalahan ini adalah *Unequal Area Facility Layout Problem* (UA-FLP) yang dikenal dengan Permasalahan Tata Letak Pabrik (TLP). Bagian ini akan menjelaskan karakteristik dari permasalahan dan model-model yang umum digunakan untuk menyelesaikan permasalahan tersebut.

Pada bagian Tujuan Instruksional Umum dijelaskan tentang harapan dari pengembangan algoritma ini. Selain itu, bagian ini juga menjelaskan tipe-tipe pengguna yang dapat memanfaatkan algoritma ini.

Pada bagian deskripsi algoritma, pengguna disajikan tentang tampilan algoritma. Pengguna juga diharapkan dapat memahami bagian-bagian algoritma secara umum. Selain itu, pengguna juga mendapatkan penjelasan tentang fitur-fitur yang dapat dimanfaatkan.

Pada bagian tahapan penggunaan algoritma dijelaskan tentang tahapan-tahapan yang harus dilakukan untuk menggunakan algoritma ini. Tahapan-tahapan ini disertai dengan langkah-langkah teknis maupun penggambaran (*screenshot*) yang dirasa perlu.

## 3. Dasar Teori

### **Teori Dasar Permasalahan Tata Letak Fasilitas dengan Luas Tak Sama (UA-FLP)**

Permasalahan tata letak fasilitas dengan luas tak sama (UA-FLP) biasanya digunakan untuk memodelkan sebuah permasalahan tata letak di sebuah fasilitas manufaktur. Permasalahan ini pertama kali diformulasikan oleh Armour and Buffa (1963). Mereka mengasumsikan ada sebuah fasilitas berbentuk persegi panjang

dengan dimensi yang tetap, panjang  $H$  dan lebar  $W$ . Kemudian ada sejumlah departemen harus disusun dan diletakkan ke dalam fasilitas tersebut. Jumlah departemen, luas tiap departemen, dan nilai aliran material tiap pasangan departemen diasumsikan diketahui. Fungsi tujuan untuk permasalahan ini adalah untuk meminimumkan biaya total perpindahan material yang diformulasikan dalam:

$$Biaya\ Total = \sum_{i=1}^n \sum_{j=1, i \neq j}^n d_{ij} f_{ij} c_{ij} \quad (1)$$

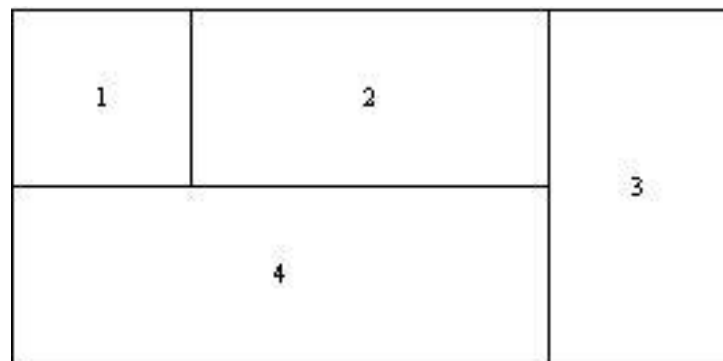
$n$  = jumlah departemen

$f_{ij}$  = Jumlah aliran material dari departemen  $i$  ke departemen  $j$ , dengan  $i, j = 1, 2, \dots, n$

$d_{ij}$  = jarak antara departemen  $i$  ke departemen  $j$ , dengan  $i, j = 1, 2, \dots, n$

$c_{ij}$  = Biaya untuk memindahkan perunit material dari departemen  $i$  ke departemen  $j$ , dengan  $i, j = 1, 2, \dots, n$

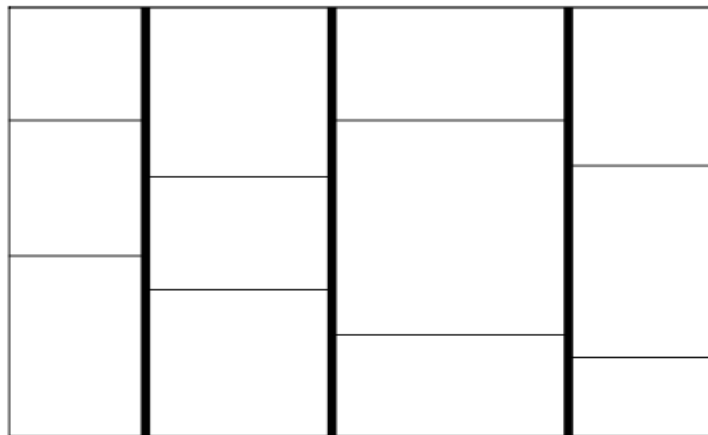
Tujuan dari permasalahan ini adalah untuk mempartisi fasilitas tersebut menjadi sub-fasilitas sehingga didapat biaya perpindahan material yang minimal. Tujuan ini didasarkan atas prinsip penanganan material (*material handling*) yakni biaya perpindahan material akan bertambah seiring dengan bertambahnya jarak yang mesti ditempuh. Gambar 1 memperlihatkan sebuah contoh solusi dari permasalahan ini, dengan panjang fasilitas 20 m dan lebar 10 m. Ada 4 departemen yang akan diletakkan ke dalam fasilitas tersebut; departemen 1 membutuhkan 25 m<sup>2</sup>, departemen 2 membutuhkan 50 m<sup>2</sup>, departemen 3 membutuhkan 50 m<sup>2</sup> dan departemen 4 membutuhkan 75 m<sup>2</sup>.



Gambar 1. Contoh solusi dari UA-FLP

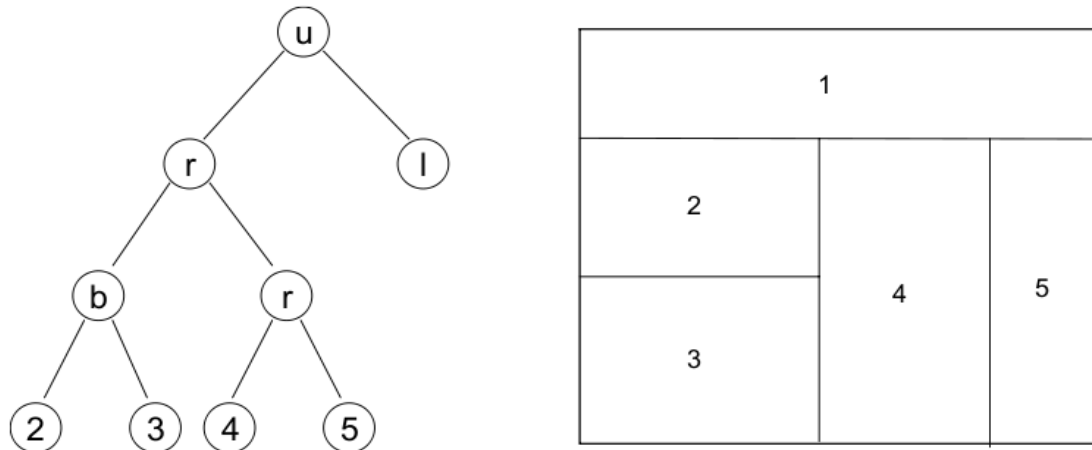
Untuk menyederhanakan permasalahan ini, para peneliti mengajukan beberapa model. Model yang sering digunakan dalam algoritma metaheuristik adalah model struktur kolom fleksibel (*Flexible Bay Structure*, FBS) dan struktur pohon pemotong (*Slicing Tree Structure*, STS).

Pada model FBS, peletakan departemen-departemen akan membentuk sebuah kolom atau baris. Tiap kolom tidak harus memiliki lebar yang sama dan tidak harus memiliki jumlah departemen yang sama. Lebar kolom secara otomatis ditentukan oleh departemen-departemen yang terkandung dalam kolom tersebut. Dengan demikian, permasalahan ini menjadi lebih sederhana untuk diselesaikan karena hanya dituntut untuk mengatur urutan departemen untuk diletakkan kedalam kolom dan menentukan jumlah departemen yang akan dikandung tiap-tiap kolom. Gambar 2 menunjukkan sebuah solusi UA-FLP yang menggunakan model FBS.



Gambar 2. Solusi UA-FLP dengan menggunakan model FBS (Shebanie, 2004)

Sedangkan pada model STS, solusi dari UA-FLP direpresentasikan oleh sebuah pohon. Representasi pohon tersebut mengandung departemen sebagai node dan juga “u”, “b”, “r”, and “l” sebagai penghubung node-node. Penghubung node memainkan peran sebagai operator pemotong (*slicing*) dalam pembentukan sub-fasilitas. Keempat penghubung node adalah “u” untuk potongan atas, “b” untuk potongan bawah, “r” untuk potongan kanan, and “l” untuk potongan kiri. Node departemen dan penghubung node menentukan lokasi relatif dan ketersinggungan dari departemen-departemen. Gambar 3 menunjukkan sebuah solusi UA-FLP yang menggunakan model STS.



Gambar 3. Solusi UA-FLP dengan menggunakan model STS (Shebanie, 2004)

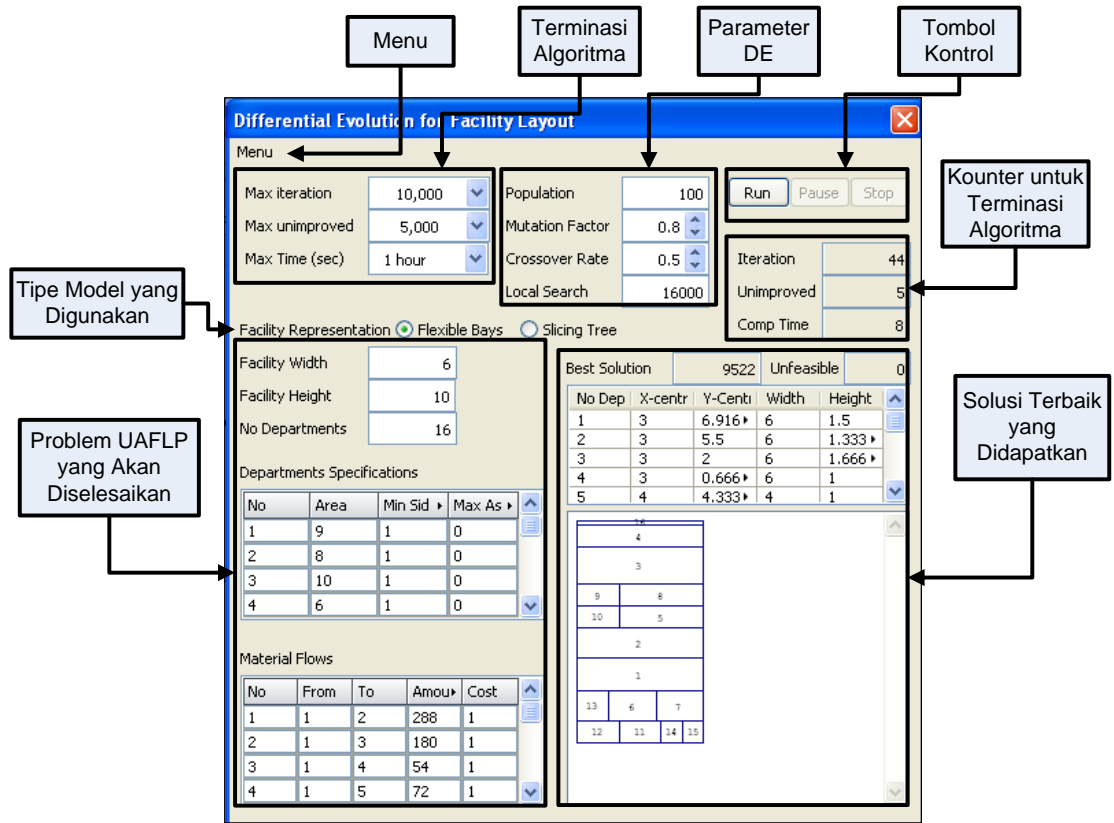
#### 4. Tujuan Instruksional

Algoritma ini bertujuan untuk membantu para pengambil keputusan dalam merancang tata letak fasilitas yang efisien dalam hal perpindahan material. Algoritma ini menyediakan beberapa pilihan solusi baik dengan model FBS maupun model STS. Selain itu, algoritma ini dapat digunakan oleh para peneliti maupun mahasiswa sebagai salah satu perbandingan metode penyelesaian UA-FLP.

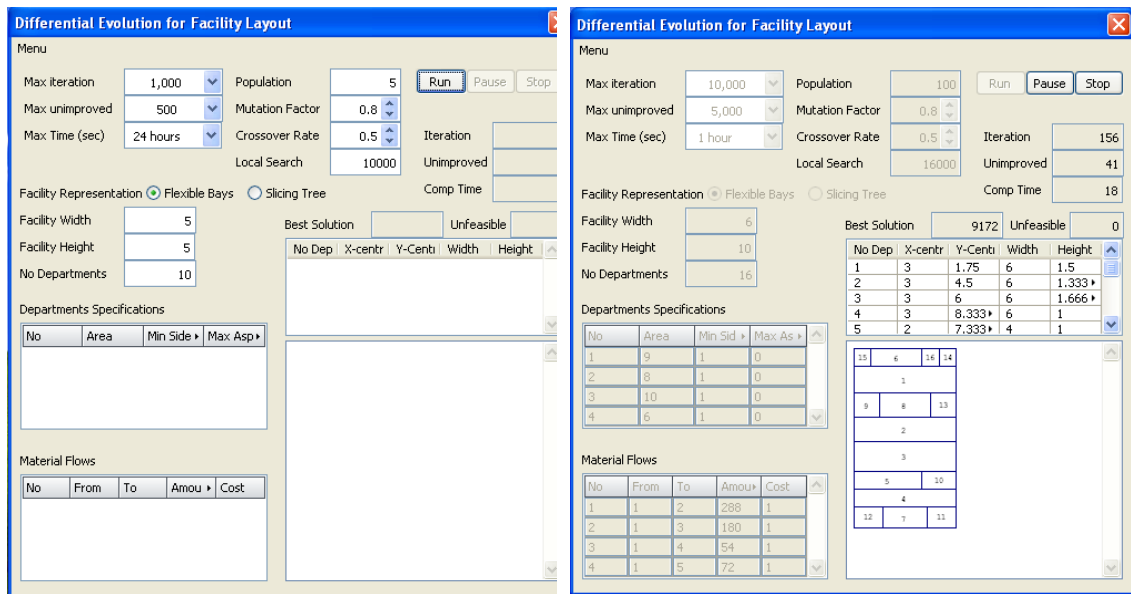
Target pengguna adalah para praktisi yang bergelut di bidang manajemen pabrik maupun manajemen rantai pasok. Selain itu, algoritma ini juga dapat digunakan untuk merancang fasilitas secara umum yang memiliki nilai keterkaitan antara departemen-departemennya sehingga algoritma ini dapat digunakan oleh perancang gedung maupun ruangan. Algoritma ini juga bisa dijadikan objek pembandingan bagi peneliti yang berlatar belakang bidang ilmu Teknik Industri, Manajemen Operasi, dan atau yang sejenisnya.

#### 5. Deskripsi Algoritma

Algoritma DE ini disertai dengan antarmuka pengguna (*user interface*) yang mudah digunakan. Gambar 4 menunjukkan tentang bagian-bagian algoritma, sedangkan Gambar 5 menunjukkan tampilan dari algoritma ini (a) tampilan awal, dan (b) saat dijalankan.



Gambar 4. Bagian-bagian algoritma



(a)

(b)

Gambar 5. Tampilan program (a) tampilan awal, dan (b) saat dijalankan.

Algoritma ini memiliki beberapa fitur yang bisa digunakan oleh pengguna, diantaranya adalah:

- Fitur input parameter dan data melalui GUI (*Graphic User Interface*).
- Fitur menyimpan dan *me-load* parameter dan data dari dan ke file.
- Fitur menyimpan hasil layout ke file.
- Fitur menyimpan gambar layout ke file.
- Fitur menjalankan, memberhentikan sementara dan menghentikan iterasi algoritma DE melalui GUI.
- Fitur meng-*update* secara otomatis hasil layout beserta gambarnya di GUI ketika ditemukan solusi baru yang lebih baik.

## 6. Tahapan Penggunaan Algoritma

Secara umum, tahapan penggunaan algoritma adalah sebagai berikut:

1. Membuka (program) algoritma.
2. Menginput Data.
3. Menjalankan iterasi.
4. Menghentikan iterasi.
5. Menyimpan hasil

### Membuka (Program) Algoritma

Membuka (program) algoritma bisa dilakukan dengan *double click* pada file program (DEFL.exe) secara langsung. Setelah di-*double click*, akan muncul tampilan program seperti pada Gambar 5(a).

### Menginput dan Meng-edit Data

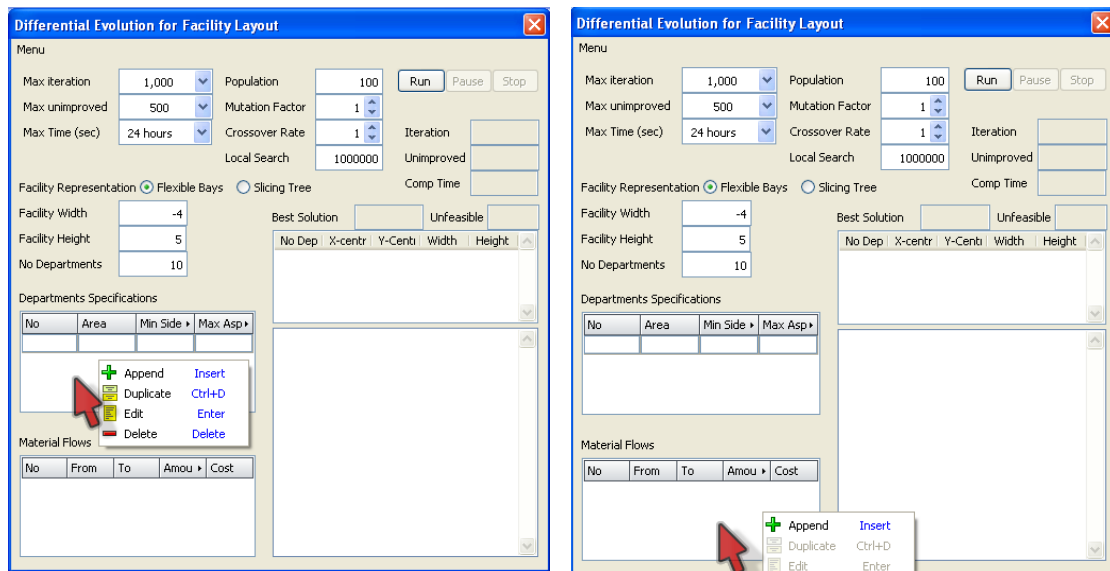
Untuk menginput data, bisa langsung dilakukan melalui GUI. Tabel 1 mendaftarkan parameter dan data yang harus diinput oleh pengguna. Tabel 1 juga menyediakan batasan-batasan nilai yang bisa diinput ke algoritma.

Tabel 1. Input data yang diperlukan

No	Input	Fungsi	Batasan
1	Max iteration	Jumlah maksimum iterasi dalam algoritma DE	Tergantung pada nilai yg tersedia pada <i>Dropdown Box</i>
2	Max unimproved	Jumlah maksimum iterasi yang tidak memberikan perbaikan terhadap solusi terbaik	Tergantung pada nilai yg tersedia pada <i>Dropdown Box</i>
3	Max time	Jumlah maksimum waktu komputasi	Tergantung pada nilai yg tersedia pada <i>Dropdown Box</i>
4	Population	Jumlah populasi vektor	Antara 5 - 100
5	Mutation Factor	Parameter yang mengatur tingkat mutasi	Antara 0 - 2
6	Crossover Rate	Parameter yang mengatur tingkat pindah silang	Antara 0 - 1
7	Local Search	Jumlah maksimum pencarian lokal yang akan dilakukan terhadap vektor trial	Antara 0 - 1.000.000
8	Facility Representation	Model penyelesaian UA-FLP	
9	Facility Width	Panjang fasilitas (sejajar dengan sumbu-x)	
10	Facility Height	Lebar fasilitas (sejajar dengan sumbu-y)	
11	No Departments	Jumlah departemen	
12	Department Specifications		
	- No	nomor departemen	
	- Area	Luas departemen	
	- Min Side	Panjang minimum dari sisi-sisi departemen yang dibutuhkan	
	- Max Aspect Ratio	Maksimum rasio antara kedua sisi departemen	
13	Material Flows		
	- No	Nomor aliran material	
	- From	Departemen sumber aliran	
	- To	Departemen tujuan aliran	
	- Amount	Jumlah unit material yang dipindahkan	
	- Cost	Biaya yang diperlukan untuk memindahkan satu unit material	

Untuk menginput data yang menyediakan *Dropdown Box*, pengguna hanya bisa memilih nilai yg tersedia di dalam *Dropdown Box* tersebut. Adapun untuk *TextBox*, data bisa langsung diketik dengan menggunakan *keyboard*. Untuk parameter '*Facility Representation*', algoritma hanya menyediakan dua pilihan yang bisa dipilih dengan meng-klik *Radio Button*-nya.

Adapun untuk data yang bersifat tabel, pengguna harus menambah baris terlebih dahulu dengan meng-klik kanan pada daerah tabel tersebut kemudian pilih *Append*. Hal ini ditunjukkan oleh Gambar 6(a) dan 6(b). Menghapus maupun menyalin baris bisa juga dilakukan dengan mengklik kanan pada daerah tabel dan memilih pilihan yang bersesuaian.



(a)

(b)

Gambar 6. Tampilan program untuk menambah baris pada (a) ‘*Department Specifications*’, dan (b) ‘*Material Flows*’.

Selain cara diatas, parameter data bisa juga dilakukan dengan me-load file yang pernah disimpan sebelumnya. Fungsi-fungsi yang berkenaan dengan ini adalah:

1. Fungsi me-load nilai parameter dan problem data dari file berformat *dea* yang telah disimpan sebelumnya ke dalam program. Fungsi ini bisa diakses melalui Menu > ‘Open Problem Data...’
2. Fungsi menyimpan nilai parameter dan problem data ke dalam file berformat *dea*. Fungsi ini bisa diakses melalui Menu > ‘Save Problem Data’.
3. Fungsi menyimpan nilai parameter dan problem data dengan nama file yang berbeda. Fungsi ini bisa diakses melalui Menu > ‘Save Problem Data As..’

### Memulai Iterasi

Untuk memulai iterasi, pengguna dapat menekan tombol ‘Run’. Setelah ditekan tombol ‘Run’, maka program secara otomatis memeriksa kesesuaian input data dengan batasan-batasan yang ada pada bagian 6.2. Selain itu, program juga memeriksa hal-berikut:

1. Jumlah total luas departemen-departemen harus sama dengan luas fasilitas

2. Jumlah baris dalam 'Department specifications' harus sama dengan nilai *TextBox* 'No departments'.
3. Jumlah baris dalam 'Material Flows' harus lebih besar sama dengan 3.

Jika semua persyaratan diatas terpenuhi, barulah algoritma akan memulai iterasinya. Ketika algoritma dijalankan, tampilan program akan seperti Gambar 5(b).

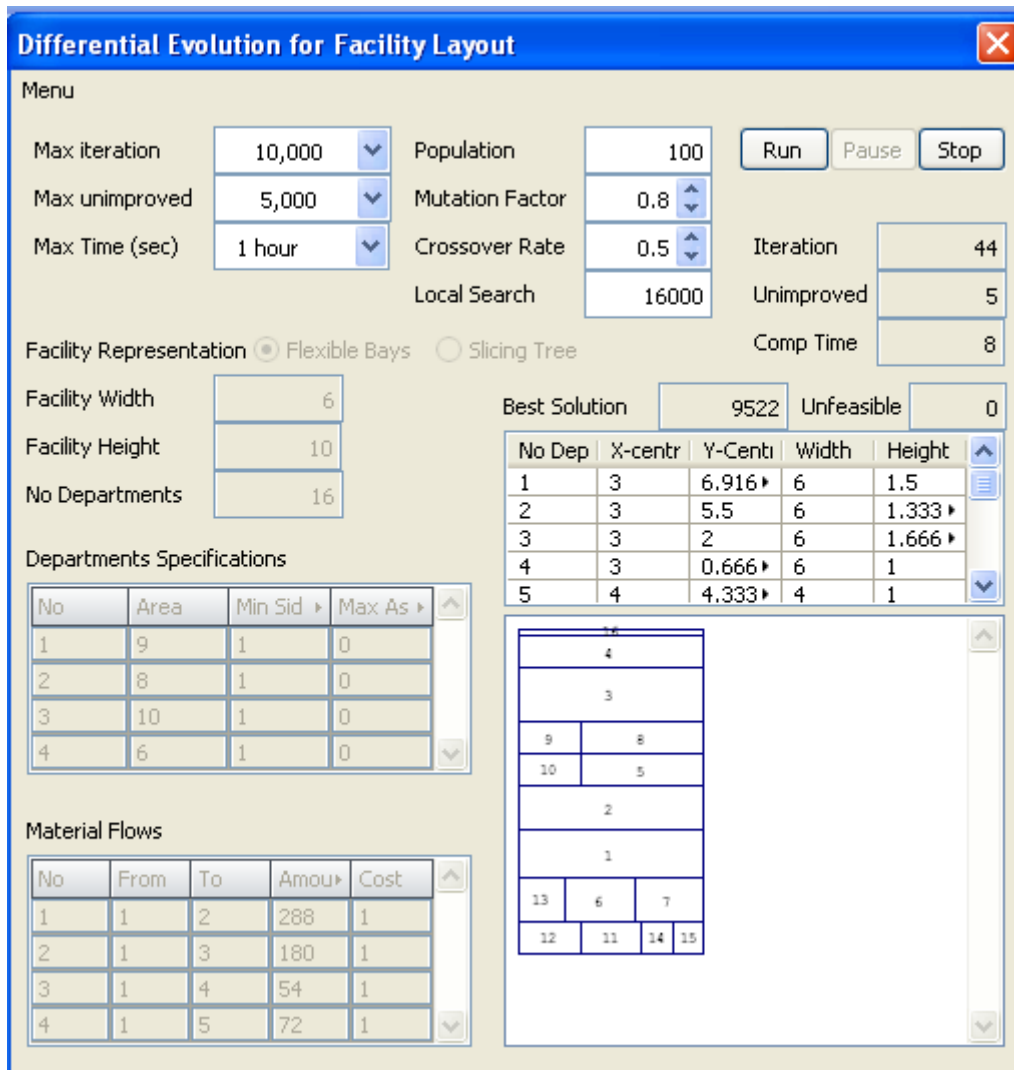
Algoritma ini juga secara otomatis meng-*update* data-data berikut:

1. Iterasi ke-*n* yang sedang dijalankan, ditampilkan pada *TextBox* 'Iteration'.
2. Jumlah iterasi yang tidak berhasil memperbaiki solusi mulai dari ditemukannya solusi global terakhir sampai iterasi saat ini, ditampilkan pada *TextBox* 'Unimproved'.
3. Jumlah waktu komputasi yang telah digunakan sampai saat ini, ditampilkan pada *TextBox* 'Comp Time'
4. Biaya terbaik (paling minimum) yang pernah ditemukan oleh algoritma sampai iterasi saat ini, ditampilkan pada *TextBox* 'Best Solution'
5. Jumlah departemen yang melanggar batasan yang ditentukan pada 'Department specifications', ditampilkan pada *TextBox* 'Unfeasible'
6. Posisi dan dimensi dari departemen-departemen pada solusi terbaik yang pernah dicapai, ditampilkan pada Tabel dibawah 'Best Solution'.
7. Penggambaran dari solusi terbaik yang pernah dicapai.

### **Menghentikan Sementara dan Menghentikan Permanen**

Algoritma ini memberikan pilihan untuk memberhentikan sementara (dapat dilanjutkan iterasinya) dan memberhentikan secara permanen. Memberhentikan sementara dilakukan dengan meng-klik tombol 'Pause'. Sedangkan memberhentikan secara permanen dapat dilakukan dengan meng-klik tombol 'Stop'. Menghentikan secara permanen ini dapat dilakukan walaupun algoritma belum menemui nilai terminasi-nya.

Ketika menghentikan algoritma secara sementara, pengguna diperkenankan untuk merubah parameter-parameter yang berhubungan dengan algoritma DE sebagaimana ditunjukkan oleh Gambar 7. Tetapi ketika algoritma dihentikan sementara, pengguna tidak dapat mengubah data problem UA-FLP.



Gambar 7. Tampilan ketika iterasi dihentikan sementara

## Menyimpan Hasil

Menyimpan hasil dapat dilakukan dengan mengakses 'Menu' sebagai berikut:

1. Menyimpan hasil tata letak akhir ke dalam file berformat *txt*. Fungsi ini bisa diakses melalui Menu > Export Result.
2. Menyimpan hasil tata letak akhir dalam bentuk gambar (*image*) ke dalam file berformat *png*. Fungsi ini bisa diakses melalui Menu > Export layout.