

Perbaikan Tata Letak Fasilitas dengan Pendekatan Simulasi dan *Evolutionary Algorithm* pada Segmen I PT X

Aloysius Fung Elnino Senatra¹, I Gede Agus Widyadana²

Abstract: PT X is a circuit breaker company that produces various types of products. There are several production segments and each segment are used to produce one product type. PT X has conducted an evaluation of the facility layout in segment I, and found the need to improve the facility layout. Shortcomings of the current layout are the arrangement of several facilities, which located outside of the available area for segment I. The arrangement of new production layout, however, should not reduce production output. Production facility re-layout is costly; therefore, simulation is used to analyze the proposed layout. Simulation approach is used to compare the initial layout against the proposed layout. The simulation model was built using Promodel 10 software, with a simulation time of 22.5 hours and replicated for 25 times. The proposed layout is derived using an evolutionary algorithm method, with adjustment in respect to some constraints from the initial layout. The proposed layout can answer the needs of segment I, because there are no facilities that exceed the specified area and result in more products in a day than the initial model.

Keywords: facility layout improvement, simulation, promodel, evolutionary algorithm

Pendahuluan

PT X adalah perusahaan pemutus arus (*circuit breaker*) yang menghasilkan berbagai jenis produk. Produk yang dihasilkan oleh PT X digolongkan berdasarkan segmen produksi tertentu, salah satunya adalah produk I pada segmen I. PT X telah melakukan evaluasi terhadap tata letak fasilitas pada segmen I, dan mendapati perlunya perbaikan tata letak fasilitas.

Segmen I telah beberapa kali melakukan perubahan tata letak fasilitas, akan tetapi perubahan hanya didasarkan pada pengalaman masa lalu. Perusahaan menginginkan adanya perbaikan pada penataan tata letak fasilitas, sebab terdapat kekurangan pada tata letak yang saat ini digunakan. Kekurangan pada tata letak saat ini adalah terdapat beberapa fasilitas yang ditempatkan di luar area yang tersedia untuk segmen I. Area yang terhalang oleh fasilitas tersebut akan digunakan untuk perbaikan serta perawatan kabel dan pipa, sehingga perbaikan tata letak perlu dilakukan. Akan tetapi, perusahaan mengkhawatirkan terjadinya perubahan pada jumlah hasil produksi sebagai dampak dari perbaikan tata letak fasilitas. Perusahaan tidak dapat mengetahui besarnya dampak dari perubahan yang akan dilakukan,

sehingga timbul kekhawatiran berkurangnya hasil produksi.

Salah satu cara yang dapat dilakukan untuk menyelesaikan persoalan tersebut adalah dengan menggunakan simulasi. Simulasi dapat menggambarkan proses dan perubahan pada suatu sistem, tanpa harus melakukan perubahan di dunia nyata. Pendekatan simulasi digunakan sebagai solusi untuk menggambarkan hasil produksi yang didapatkan dari tata letak baru dan lama. Jumlah hasil produksi yang didapatkan dari simulasi, akan dibandingkan dan digunakan sebagai dasar pertimbangan kelayakan perbaikan tata letak. Tata letak yang baru diharapkan dapat menempatkan fasilitas sesuai batas area segmen I, serta memberikan hasil produksi yang lebih banyak.

Metode Penelitian

Penelitian ini menggunakan beberapa metode yang berkaitan dengan penggunaan simulasi dalam perbaikan tata letak fasilitas. Tata letak fasilitas saat ini akan disimulasikan dan dibandingkan dengan simulasi tata letak fasilitas usulan yang dibuat menggunakan *evolutionary algorithm*.

Tata Letak Fasilitas

Menurut Stephens dan Meyers [1], tata letak fasilitas merupakan merupakan pengorganisasian aset fisik

^{1,2} Fakultas Teknologi Industri, Jurusan Teknik Industri, Universitas Kristen Petra. Jl. Siwalankerto 121-131, Surabaya 60236. Email: fungalloysius@gmail.com, gede@petra.ac.id

perusahaan untuk meningkatkan efisiensi sumber daya seperti pekerja, material, peralatan, dan energi. Pengaturan tata letak fasilitas mempertimbangkan luas area yang tersedia untuk menempatkan mesin, mengatur kelancaran perpindahan material, penempatan pekerja, dan lain sebagainya. Tujuan dari penataan letak fasilitas menurut Wignjosuebrotto [2] adalah:

- Meminimalkan transportasi dan proses pemindahan barang.
- Meminimalkan gerakan balik yang tidak diperlukan.
- Meminimalkan penggunaan tanah.
- Memberikan pola aliran produksi terbaik.
- Menyeimbangkan lintasan perakitan.
- Memberikan fleksibilitas untuk mempersiapkan kemungkinan ekspansi.

Terdapat berbagai pendekatan yang dapat digunakan dalam perbaikan tata letak, salah satunya adalah melalui pendekatan simulasi. Perbaikan tata letak melalui simulasi bukanlah hal baru, karena telah dilakukan dalam penelitian-penelitian sebelumnya oleh Jayachitra dan Prasad [3], Garcia *et al.* [4], serta Smutkupt dan Wimonkasame [5].

Evolutionary Algorithm

Pendekatan lain yang dapat digunakan dalam perbaikan tata letak fasilitas adalah *evolutionary algorithm*. Sama seperti pendekatan simulasi, perbaikan tata letak melalui *evolutionary algorithm* telah dilakukan dalam penelitian-penelitian sebelumnya oleh Schnecke dan Vornberger [6], Norman, Smith, dan Arapoglu [7], serta Norman *et al.* [8].

Algoritma evolusioner (EA) merupakan algoritma yang dapat melakukan optimasi atau menjalankan perintah untuk belajar, dengan kemampuan untuk berkembang. Menurut Yu dan Gen [9], terdapat tiga karakteristik utama dari algoritma evolusioner, yaitu:

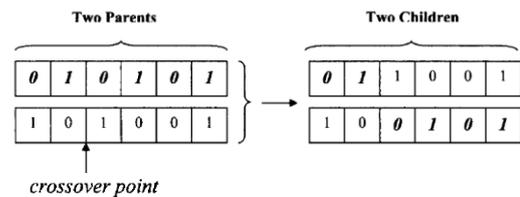
- Berbasis populasi.
EA menyimpan sekelompok solusi, yang dinamakan populasi, untuk melakukan optimasi atau mempelajari suatu masalah secara paralel.
- Berorientasi pada fitness.
Setiap solusi dalam populasi disebut suatu individu. Setiap individu memiliki representasi gen-nya, yang bernama kode (*code*), dan evaluasi performa, yang bernama *fitness value*. EA lebih mengutamakan individu dengan nilai fit yang lebih baik, sebagai dasar optimasi dan konvergensi dari algoritma tersebut.

- Tergerak oleh variasi.
Masing-masing individu akan mengalami berbagai perubahan melalui variasi untuk menirukan perubahan genetik, yang penting untuk mencari daerah solusi.

Variasi dari suatu individu didapatkan dengan pertukaran informasi, melalui operator variasi. Menurut Simon [10], pertukaran informasi dapat dilakukan melalui persilangan (*crossover*), atau juga dapat disebut rekombinasi (*recombination*), antara dua atau lebih individu, serta melalui mutasi.

Persilangan

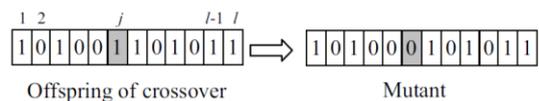
Persilangan dilakukan dengan mengawinkan dua individu induk untuk menghasilkan dua keturunan. Perkawinan individu merupakan proses penyilangan *allele*, sehingga didapatkan keturunan yang baru. Individu disilangkan dengan menentukan titik persilangan (*crossover point*) berdasarkan nilai acak. Persilangan dapat digambarkan melalui Gambar 1.



Gambar 1. Ilustrasi persilangan dua individu
Sumber: Simon [10]

Dua individu baru dihasilkan dengan menukarkan nilai *allele* setelah titik persilangan. Kedua induk mati dan digantikan oleh keturunan yang baru, sehingga menghasilkan satu generasi. Individu-individu baru ini juga memiliki nilai *fitness*, dan akan dipasangkan satu dengan lainnya.

Mutasi dilakukan dengan mengganti nilai *allele* pada suatu gen, dengan angka kebalikannya (*bit-flip mutation*). Setiap individu memiliki probabilitas untuk bermutasi, yang dinyatakan oleh *mutation rate*. Tahapan mutasi diilustrasikan pada Gambar 2.



Gambar 2. Mutasi *bit-flip* pada gen ke-*j* dari keturunan
Sumber: Yu dan Gen [9]

Mutasi *bit-flip* dilakukan dengan merubah nilai *allele* dari gen ke-*j* dengan angka kebalikannya. Keturunan yang telah mengalami perubahan nilai, akan disebut sebagai mutan.

Hasil dan Pembahasan

Data yang digunakan dalam penelitian ini adalah peta proses operasi, denah serta ukuran tata letak fasilitas, sampel waktu proses operasi, bahan baku yang digunakan pada bulan Agustus hingga September 2019, serta hasil produksi bulan Agustus hingga September 2019. Pengolahan data dilakukan untuk memodelkan proses produksi berdasarkan tata letak fasilitas awal.

Uji Identik dan Independen

Data waktu proses haruslah independen dan identik, sehingga perlu dilakukan pengujian sebelum dapat digunakan sebagai data waktu proses simulasi. Uji independen dilakukan pada seluruh data dengan menggunakan uji *autocorrelation*, dan digambarkan melalui *scatterplot* jika terdapat *lag* yang melebihi batas. Uji identik dilakukan jika data diambil pada waktu yang berbeda, menggunakan *test of equal variance*, dengan α sebesar 0,05. Tidak semua data dapat diuji identik, karena keterbatasan waktu. Hasil dari uji independen dan uji identik terhadap data waktu proses produksi, dapat dilihat pada Tabel 1. Simbol Y berarti lulus uji, simbol T berarti tidak diuji, sedangkan simbol X berarti gagal uji.

Tabel 1. Hasil uji identik dan independen

Nama data	Identik	Independen
<i>Packing manual</i>	Y	Y
<i>Packing</i>	T	Y
<i>Loading</i>	T	Y
<i>Press kontakt bimetal</i>	Y	Y
<i>Press kontakt sockel</i>	Y	Y
<i>Winding draht</i>	Y	Y
<i>Abisolieren</i>	Y	Y
Prates	T	Y
<i>Welding draht bimetal</i>	T	Y
<i>Welding bimetal anschluss</i>	T	Y
<i>Welding draht anschluss</i>	Y	Y
<i>Press kappe</i>	T	Y
<i>Tampoprint</i>	T	Y
<i>Drying bimetal</i>	Y	Y
Mesin TMC	T	Y
<i>Tes kontakt</i>	T	Y

Uji Distribusi

Uji distribusi dilakukan dengan membuat dugaan awal terhadap bentuk distribusi data. Bentuk distribusi dapat diketahui melalui sebaran data dalam bentuk histogram. Uji kesesuaian distribusi dilakukan menggunakan *stat:fit* pada perangkat lunak *Promodel* dengan α sebesar 0,05, dan hipotesa awal data berdistribusi sesuai dugaan. Metode yang digunakan dalam melakukan uji distribusi adalah uji

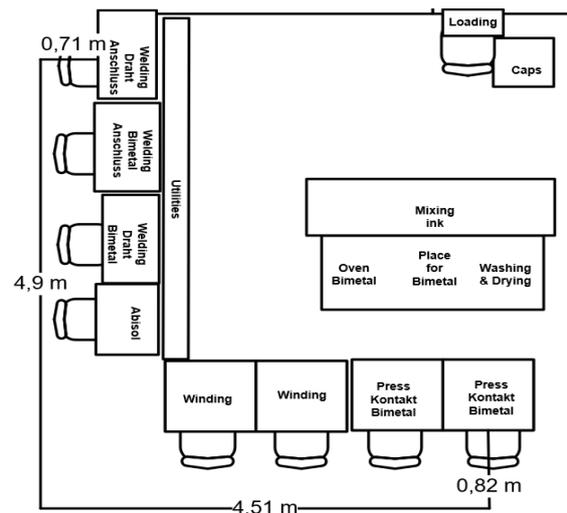
kolmogorov-smirnov untuk data berjumlah di bawah 30, dan uji *chi-square* untuk data berjumlah lebih dari 30. Distribusi waktu dari seluruh proses produksi dapat dilihat pada Tabel 2.

Tabel 2. Distribusi seluruh waktu proses produksi

Nama data	Bentuk Distribusi
<i>Packing manual</i>	Lognormal(3.55, 0.43)
<i>Packing</i>	Beta(0.738892, 3.53743, 2.12, 3.61)
<i>Loading</i>	Lognormal(3.2, 0.158)
<i>Press kontakt bimetal</i>	Gamma(4.81226, 0.197172)
<i>Press kontakt sockel</i>	Gamma(1.71554, 0.459552)
<i>Winding draht</i>	Beta(2.3889, 7.57317, 9.41, 15.6492)
<i>Abisolieren</i>	Lognormal(5.94, 1.15)
Prates	Gamma(170.236, 0.120574)
<i>Welding draht bimetal</i>	Weibull(2.98468, 1.83909)
<i>Welding bimetal anschluss</i>	Lognormal(5.83, 0.926)
<i>Welding draht anschluss</i>	Weibull(1.75047, 1.5775)
<i>Press kappe</i>	Lognormal(2.85, 0.561)
<i>Tampoprint</i>	Lognormal(2.22, 0.0962)
<i>Drying bimetal</i>	Weibull(3.40574, 65.3546)
Mesin TMC	Lognormal(215, 8.57)
<i>Tes kontakt</i>	Lognormal(1.77, 0.345)

Pengolahan Data Tata Letak Fasilitas

Data yang diperlukan untuk membuat tata letak fasilitas awal adalah *flow* dan jarak antar fasilitas. Nilai *flow* didapatkan melalui jumlah hasil produksi masing-masing fasilitas berdasarkan model awal. Perhitungan jarak antar fasilitas dapat dilihat pada Gambar 3.



Gambar 3. Perhitungan jarak antar fasilitas

Perhitungan jarak dilakukan berdasarkan jalur perpindahan material yang digunakan. Ukuran fasilitas dapat dilihat pada Tabel 3.

Tabel 3. Ukuran seluruh fasilitas

No	Nama Fasilitas	Ukuran (m)
1	Rak Finished Goods	1,60 x 0,60
2	Oven & Washing	2,46 x 0,84
3	Press Kontakt Bimetal 1	0,90 x 1,235
4	Press Kontakt Bimetal 2	0,90 x 1,235
5	Winding 1	0,90 x 1,235
6	Winding 2	0,90 x 1,235
7	Winding 3	0,90 x 1,235
8	Winding 4	0,90 x 1,235
9	Winding M1	0,90 x 1,235
10	Winding M2	0,90 x 1,235
11	Abisol 1	1,00 x 0,80
12	Abisol 2	1,00 x 0,80
13	Abisol M	0,93 x 0,62
14	Welding DB1	1,10 x 1,00
15	Welding DB2	1,10 x 1,00
16	Welding DBM	0,945 x 1,00
17	Welding BA1	1,05 x 1,00
18	Welding BA2	1,05 x 1,00
19	Welding BAM	0,945 x 1,00
20	Welding DA1	1,00 x 1,00
21	Welding DA2	1,00 x 1,00
22	Welding DAM	0,945 x 1,00
23	Press Kontakt Sockel 1	1,10 x 1,20
24	Press Kontakt Sockel 2	0,625 x 1,125
25	Rak Siap Pra Uji	0,81 x 0,41
26	Pra Uji	1,45 x 1,56
27	Rak Loyang Pra Tes	0,81 x 0,41
28	Press Kappe	1,75 x 1,00
29	Tampo	1,535 x 0,66
30	Packing Manual	1,80 x 1,16
31	Loading	0,80 x 0,67
32	Mesin TMC	5,25 x 3,20
33	Unloading	0,80 x 0,67
34	Ruang Tes	1,025 x 0,677
35	Rak Curing	2,85 x 0,60
36	Troli Sockel	8,00 x 6,50

Pembuatan Model

Model dibuat menggunakan perangkat lunak *Promodel*. Terdapat beberapa batasan yang digunakan dalam pembuatan model, yaitu:

- Entitas yang dimodelkan adalah *bimetal* dan *sockel*.
- Entitas pendukung, seperti *kappe*, keling, kawat, dan cat tidak dimodelkan.
- Setiap mesin memiliki satu pekerja.
- Waktu proses pada operasi *curing* adalah satu detik.

Penyederhanaan model simulasi dilakukan dengan menempatkan satu pekerja pada setiap proses, serta hanya memodelkan entitas *bimetal* dan *sockel*. Batasan waktu proses *curing* ditetapkan untuk menghindari ketidakakuratan analisis yang dihasilkan *output viewer*. Ketidakakuratan terjadi karena proses *curing* menambahkan waktu *idle* selama 24 jam pada seluruh proses produksi.

Uji Kecukupan Data

Model dijalankan selama 22,5 jam, sesuai dengan waktu produksi dalam sehari. Waktu *warmup time* yang digunakan sebesar 160 menit. Replikasi dilakukan sebanyak 25 kali, dan akan diuji kelayakannya melalui uji kecukupan replikasi. Uji replikasi dilakukan dengan menggunakan Rumus 1.

$$n' = \left(\frac{s \times t}{k \times \bar{x}} \right)^2 ; \quad n \leq n' \quad (1)$$

Pengujian dilakukan terhadap 25 data hasil simulasi, dengan rata-rata hasil produksi sebanyak 26.935,32 buah dan standar deviasi 56,21 buah. Hasil uji kecukupan data adalah, diperlukan sebanyak 0,009958776 kali replikasi. Jumlah replikasi yang digunakan lebih besar dari yang diperlukan, sehingga replikasi model awal telah mencukupi.

Verifikasi Model

Tahap verifikasi bertujuan untuk memeriksa logika operasional suatu model. Verifikasi dilakukan dengan mengubah lama waktu operasi, lalu membandingkan rata-rata waktu entitas di dalam sistem. Model terverifikasi jika terdapat perubahan rata-rata waktu entitas di dalam sistem.

Verifikasi model menunjukkan adanya peningkatan rata-rata waktu dalam sistem, ketika waktu proses ditambahkan. Rata-rata waktu dalam sistem juga turut berkurang ketika waktu proses dikurangi. Didapati bahwa model telah terverifikasi, seperti yang terlihat pada Tabel 4.

Tabel 4. Data hasil verifikasi model awal

No	Nama Entitas	Waktu Pengerjaan (detik)	Average Time in System
1	Bimetal	60 (awal)	888.78
		180	938.8
		0	821.67
2	Sockel	60 (awal)	878.91
		180	929.09
		0	793.41

Validasi Model

Uji validasi dilakukan menggunakan uji *two-sample t* pada perangkat lunak *minitab*. Uji *two-sample t* dilakukan menggunakan nilai *a* sebesar 0,05, dan hipotesa nol tidak terdapat perbedaan yang signifikan antara hasil produksi dan model simulasi. Hasil uji *two-sample t* dapat dilihat pada Gambar 4.

Test

Null hypothesis	$H_0: \mu_1 - \mu_2 = 0$	
Alternative hypothesis	$H_1: \mu_1 - \mu_2 \neq 0$	
T-Value	DF	P-Value
0.73	63	0.465

Gambar 4. Tampilan hasil uji *two-sample t* model awal

Nilai *p-value* yang lebih besar dari nilai α , menunjukkan gagal tolaknya hipotesa awal. Dapat disimpulkan bahwa tidak terdapat perbedaan yang signifikan antara hasil yang didapatkan dari model simulasi, terhadap hasil produksi.

Penggunaan *Evolutionary Algorithm*

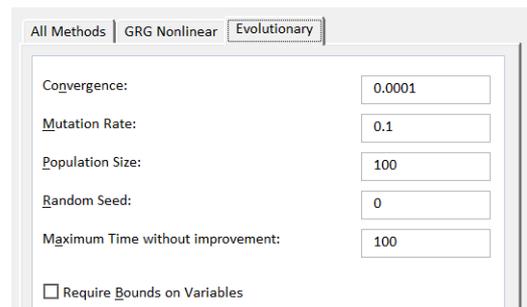
Model usulan dibuat berdasarkan hasil dari model awal. Data *output* antar lokasi pada model awal, digunakan sebagai frekuensi perpindahan antar fasilitas. Frekuensi perpindahan serta jarak antar fasilitas diolah menggunakan *evolutionary algorithm*, untuk menghasilkan tata letak fasilitas usulan. Terdapat tiga parameter yang perlu dinyatakan untuk menjalankan *solver*, yaitu fungsi objektif, variabel, dan kendala. Fungsi objektif dapat dinyatakan melalui fungsi *sumproduct* dari matriks jarak *solver* dan *matriks flow*.

Fungsi *sumproduct* merupakan fungsi untuk menjumlah hasil perkalian antara *flow* dan jarak antar fasilitas. Parameter *array* serta fungsi dari *sumproduct*, dapat diisi seperti pada Gambar 6. Pada contoh ini, *array1* adalah isi dari matriks jarak *solver* dan *array2* adalah isi dari matriks *flow solver*.

Parameter *solver* lainnya adalah variabel pengganti, yang berguna untuk mengubah nilai *array* dari matriks jarak *solver* dan matriks *flow solver*. Nilai isi dari matriks jarak dan *flow solver*, bergantung dengan nilai indeks yang dimilikinya. Indeks baris dan kolom matriks jarak dan *flow solver* disamakan dengan variabel pengganti, seperti yang dicontohkan pada Gambar 7.

Setelah menyatakan parameter yang ada pada lembar kerja *excel*, parameter-parameter tersebut dinyatakan pada *solver parameters*. *Set objective* diisi dengan *cell* yang berisi fungsi tujuan, dengan meminimalkan fungsi tujuan. *Changing variable cells* diisi dengan *cell* yang berisikan variabel pengganti. *Constraints* yang digunakan adalah variabel pengganti dengan kendala *alldifferent*. *Solving method* yang digunakan adalah *evolutionary*.

Terdapat beberapa parameter yang dapat ditetapkan dalam penggunaan *evolutionary algorithm*. Parameter yang ditetapkan dalam perbaikan tata letak fasilitas, dapat dilihat pada Gambar 5.



Gambar 5. Parameter *evolutionary algorithm* pada *solver*

Matriks Jarak					
i/j	1	2	3	4	5
1	-	1	2	3	4
2	1	-	4	3	2
3	2	4	-	2	1
4	3	3	2	-	1
5	4	2	1	1	-

Matriks Flow					
i/j	1	2	3	4	5
1	-	5	6	7	8
2	5	-	8	8	10
3	6	8	-	9	5
4	7	8	9	-	6
5	8	10	5	6	-

Matriks Jarak Solver					
i/j	1	2	3	4	5
1	-				
2	1	-			
3	2	4			
4	3	3	2	-	1
5	4	2	1	1	-

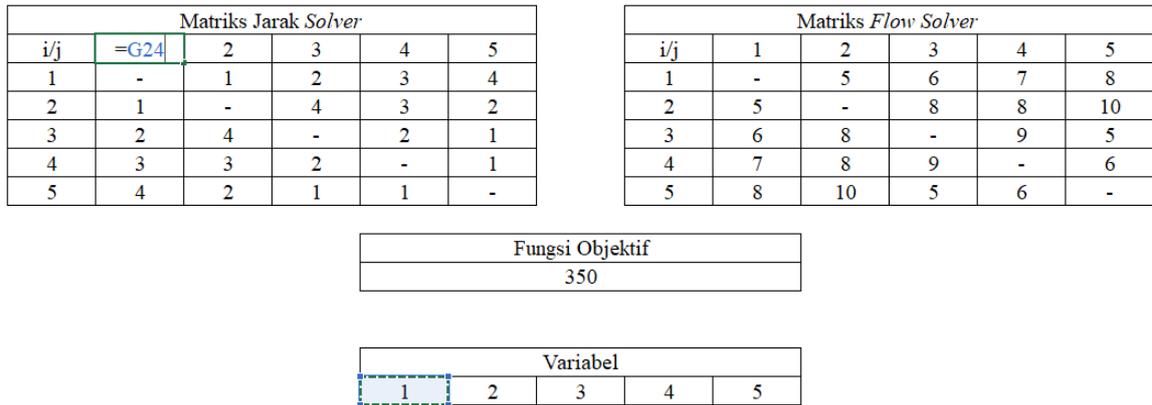
Matriks Flow Solver					
i/j	1	2	3	4	5
1	-	5	6	7	8
2	5	-	8	8	10
3	6	8	-	9	5
4	7	8	9	-	6
5	8	10	5	6	-

Fungsi Objektif

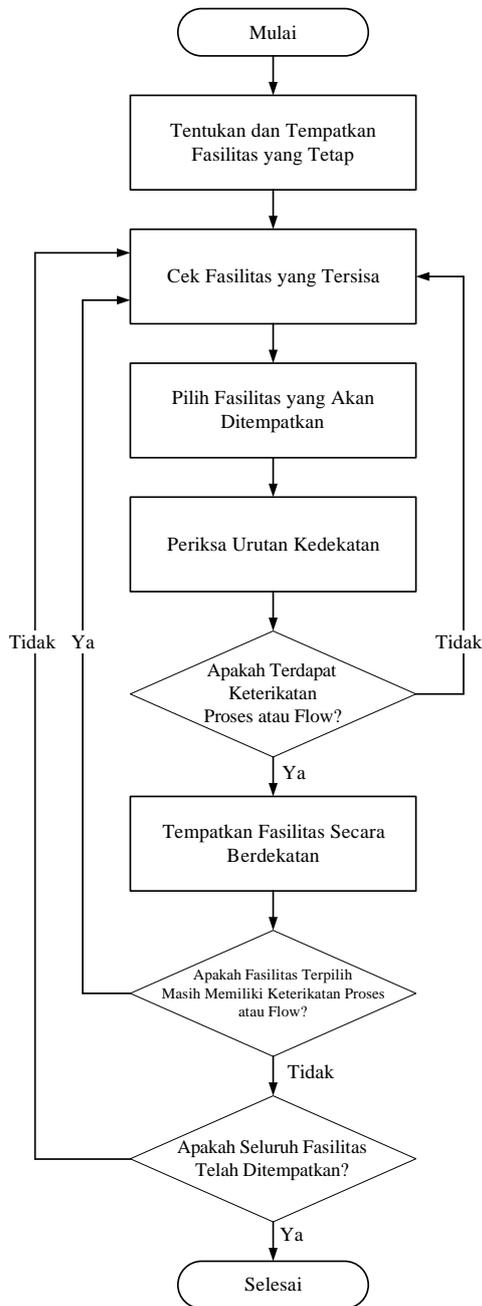
=SUMPRODUCT(D13:H17,K13:O17)

SUMPRODUCT(array1, [array2], [array3], [array4], ...)

Gambar 6. Tampilan penggunaan fungsi *sumproduct* dan indeks



Gambar 7. Tampilan penggunaan variabel pengganti



Gambar 8. Flowchart penempatan tata letak fasilitas usulan

Urutan kedekatan fasilitas yang didapatkan haruslah diolah, sebelum menghasilkan tata letak fasilitas secara utuh. Penempatan tata letak fasilitas haruslah dilakukan secara terstruktur dan konsisten, agar dapat memberikan hasil yang lebih optimal. Langkah-langkah penempatan tata letak fasilitas yang dilakukan, dapat digambarkan melalui *flowchart* seperti pada Gambar 8. Tata letak fasilitas awal dan usulan dapat dilihat pada Gambar 9 dan Gambar 10.

Pembuatan Model Usulan

Model simulasi usulan dibuat berdasarkan model awal yang telah ada. Perbedaan terdapat pada lokasi *weldingda1* dan *weldingda2*. *Weldingda1* berpindah posisi pada konveyor 2 dan *weldingda2* berpindah pada konveyor 1, sehingga terjadi perubahan pada *operations* dan *routing*. *Weldingda1* mengirimkan *bimetal* ke *bkonveyor2*, sementara *weldingda2* mengirimkan *bimetal* ke *bkonveyor1*. Batas perubahan variabel *var1* pada *bkonveyor2* menjadi 250, dan 200 untuk *bkonveyor1*.

Verifikasi Model Usulan

Verifikasi model usulan memiliki tujuan yang sama dengan verifikasi model awal. Hasil verifikasi model dapat dilihat pada Tabel 5.

Tabel 5. Data hasil verifikasi model usulan

No	Nama Entitas	Waktu Pengerjaan (detik)	Average Time in System
1	Bimetal	60 (awal)	888.81
		180	939.07
		0	821.43
2	Socket	60 (awal)	878.92
		180	929.09
		0	791.56

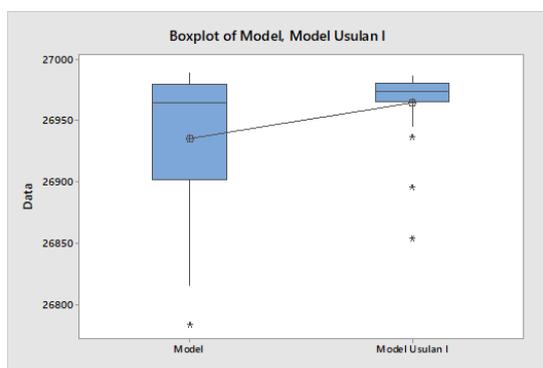
Terdapat perbedaan pada rata-rata waktu dalam sistem, sehingga model terverifikasi.

Test

Null hypothesis	$H_0: \mu_1 - \mu_2 = 0$
Alternative hypothesis	$H_1: \mu_1 - \mu_2 \neq 0$
T-Value	DF
-2.29	36
P-Value	0.028

Gambar 12. Tampilan hasil *two-sample t* model awal dan model usulan

Nilai *p-value* yang lebih kecil dari nilai α , memiliki arti tolak hipotesa awal. Hipotesa yang diterima dari uji *two-sample t* adalah, terdapat perbedaan yang signifikan antara hasil produksi model awal dan model usulan. Perbandingan hasil produksi antara model awal dan model usulan, juga dapat dilihat melalui *boxplot* pada Gambar 11.



Gambar 13. *Boxplot* model awal dan model usulan

Model awal menghasilkan variasi yang lebih besar, jika dibandingkan dengan model usulan. Variasi ini dapat dilihat melalui besarnya rentang batas minimum dan maksimum dari *boxplot*. Model usulan juga memiliki rata-rata yang lebih baik, jika dibandingkan terhadap model awal.

Hasil dari uji *two-sample t* dan *boxplot* menunjukkan adanya perbedaan yang signifikan antara model awal dan usulan. Meskipun hanya memiliki perbedaan hasil produksi sebesar 30 buah, model usulan terbukti memiliki perbedaan hasil yang signifikan. Perbedaan lain yang dapat diamati adalah standar deviasi model usulan yang lebih kecil dari model awal.

Percobaan tambahan dilakukan terhadap model usulan, dengan meningkatkan jumlah entitas yang diproses. Jumlah entitas *bimetal* dan *socket* diubah menjadi 90.000 unit per entitas. Efek dari perubahan ini adalah meningkatnya rata-rata hasil produksi menjadi 33.817 unit barang jadi. Percobaan ini menandakan kapasitas produksi maksimal dari model usulan, adalah sebesar 33.817 unit.

Simpulan

Hasil dari uji *two-sample t* dan *boxplot* menunjukkan adanya perbedaan yang signifikan antara model awal dan usulan. Meskipun hanya memiliki perbedaan hasil produksi sebesar 30 buah, model usulan terbukti memiliki perbedaan hasil yang signifikan. Perbedaan lain yang dapat diamati adalah standar deviasi model usulan yang lebih kecil dari model awal. Tata letak fasilitas usulan juga tidak memiliki fasilitas yang melebihi area kerja segmen I, sehingga dapat disimpulkan bahwa tata letak fasilitas usulan mampu menjawab tujuan penelitian.

Daftar Pustaka

- Stephens, M., and Meyers, F. *Manufacturing facilities design and material handling* (5th ed.). West Lafayette, Indiana: Purdue University Press, 2013.
- Wignjosebroto, S. *Ergonomi studi gerak dan waktu: Studi gerak dan waktu* (4th ed.). Surabaya: Penerbit Guna Widya, 2006.
- Jayachitra, R., and Prasad, P. S. S. Design and selection of facility layout using simulation and design of experiments. *Indian Journal of Science and Technology*, 3(4), 2010, pp. 437-446.
- Garcia, E. F., Zúñiga, E. R., Bruch, J., Moris, M. U., and Syberfeldt, A. Simulation-based Optimization for Facility Layout Design in Conditions of High Uncertainty. *Procedia CIRP*, 72, 2018, pp. 334-339.
- Smutkupt, U., and Wimonkasame, S. Plant layout design with simulation. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2009, Vol. 2, pp. 18-20.
- Schnecke, V., and Vornberger, O. Hybrid genetic algorithms for constrained placement problems. *IEEE Transactions on Evolutionary Computation*, 1(4), 1997, pp. 266-277.
- Norman, B. A., Smith, A. E., and Arapoglu, R. A. Integrated facility design using an evolutionary approach with a subordinate network algorithm. In *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, September 1998, pp. 937-946.
- Norman, B. A., Smith, A. E., Yildirim, E., and Tharmmaphornphilas, W. An evolutionary approach to incorporating intradepartmental flow into facilities design. *Advances in Engineering Software*, 32(6), 2001, pp. 443-453.
- Yu, X., and Gen, M. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.
- Simon, D. *Evolutionary optimization algorithms*. Hoboken, NJ: John Wiley & Sons Inc, 2013.