OPTIMASI KESEIMBANGAN LINTASAN PERAKITAN *TWO-SIDED* MENGGUNAKAN ALGORITMA *SIMULATED ANNEALING*

Riri Damayanti Apnena[1)], Abdul Syukur[2)], Guruh Fajar Shidik[3)]
Politeknik TEDC Bandung[1)], Universitas Dian Nuswantoro[1) 3)]
Email: riri.damayanti.apnena@poltektedc.ac.id[1)]

Abstract
Two-sided line is line series consists of the left side and right are dealing simultaneously. Elements of work to be done the left or right of the transport system (conveyors) and there are also elements of work that can be done on both sides and presented to the assembly of products has a high complexity (eg: product automobiles). Algorithm simulated annealing including the algorithm metaheuristic, ie, an algorithm that allows reception of the solution space larger because considering all the solutions that emerge though not a better solution and take advantage of structural changes occurring in the neighborhood to address issues if the search solution trapped in local minimum. Simulated annealing consists of phase generation solutions initials using the algorithm J-Wagon and stage repair solutions initials in search of local search (exchange and insert).

Keywords: *assembly line balancing, two-sided assembly line balancing, metaheuristic, simulated annealing.*

I. PENDAHULUAN

Latar Berlakang

Keseimbangan lintasan perakitan (*assembly line balancing*, ALB) dapat didefinisikan sebagai penugasan satu set elemen kerja N_j (j = 1, 2, 3, ..., n) ke dalam satu set stasiun kerja M_i (i = 1, 2, 3, ..., m) yang serial dalam suatu lintasan produksi sehingga setiap stasiun kerja memiliki waktu stasiun yang tidak melebihi waktu siklus dari lintasan tersebut. Penugasan-penugasan tersebut dilakukan tanpa melanggar *precedence relationship* untuk memenuhi satu atau beberapa fungsi tujuan, seperti minimisasi jumlah stasiun kerja atau maksimasi lintasan perakitan.

Pada kondisi nyata, penggunaan lintasan perakitan tunggal (SALB) dalam suatu industri menjadi tidak efisien ketika dihadapkan pada perakitan produk-produk yang memiliki kompleksitas tinggi. Untuk mengatasi masalah keseimbangan lintasan perakitan tersebut, dikembangkanlah suatu lintasan perakitan dua sisi *two sided assembly line balancing problem* (TALBP). TALBP pertama kali dipelajari oleh Bartholdi dengan menggambarkan desain dan penggunaan sebuah program komputer interaktif berbasis aturan prioritas sederhana untuk lintasan perakitan dua sisi, contoh produknya truk [2].

Simulated annealing (SA) merupakan salah satu algoritma metaheuristik yang mengadopsi proses pendinginan logam yang mengharuskan perubahan secara perlahan untuk mendapatkan karakteristik (paduan) logam yang baik. Pada SA, algoritma membolehkan menerima solusi yang tidak lebih baik dengan probabiltias tertentu. Probabilitas ini akan menurun seiring berjalannya iterasi.

Berdasarkan penelitian-penelitian sebelumnya, terlihat bahwa algoritma SA belum diterapkan untuk menyelesaikan TALB. Oleh karena itu, penelitian ini akan mengembangkan model TALB menggunakan algoritma *simulated annealing* dengan kriteria minimisasi jumlah stasiun kerja. Diharapkan dengan melakukan pengembangan algoritma SA pada TALB dapat efisiensi lintasan sama dengan peneltian sebelumnya dan menurunkan nilai *smoothness index* dari lintasannya.

Rumusan masalah pada penelitian ini, keseimbangan lintasan perakitan dua sisi, yakni pendistribusian banyaknya elemen kerja pada stasiun kerja tanpa melanggar posisi pada precedence diagram atau waktu tiap stasiun kerja tidak melebihi waktu siklus dengan kriteria minimisasi jumlah stasiun kerja. Oleh karena itu, agar menghasilkan jumlah stasiun kerja yang minimum, yaitu efisisensi yang tinggi (minimal sama dengan penelitian sebelumnya) serta *smoothness index* (diharapkan lebih kecil dari penelitian sebelumnya) maka diperlukan algoritma *simulated annealing*.

Penelitian ini bertujuan memberikan solusi pada masalah TALB dengan menggunakan algoritma SA untuk minimisasi jumlah stasiun kerja didukung dengan perangkat lunak. Solusi

yang didapat adalah data penggelompokkan elemen kerja pada suatu stasiun kerja, jumlah stasiun kerja yang dihasilkan, efisiensi lintasan dan *smoothness index* dari lintasan yang dibentuk.

Manfaat yang diberikan pada penelitian ini adalah

1. Manfaat bagi masyarakat
   Dapat mempermudah menentukan jumlah stasiun kerja yang optimal pada TALB.
2. Manfaat pengembangan iptek
   Dapat memberikan pengembangan dari algoritma *simulated annealing* pada masalah TALB.
3. Manfaat bagi peningkatan nilai ekonomi/perusahaan
   Dapat menjadi pertimbangan untuk menentukan model keseimbangan lintasan perakitan dua sisi pada suatu perusahaan yang bergerak dibidang otomobil.

II. TINJUAUAN PUSTAKA

Penelitian Terkait

Ringkasan hasil dari penelitian-penelitian yang relevan dapat dilihat pada Tabel 1.

Tabel 1. Ringkasan hasil dari penelitian yang relevan

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1	Kim *et al.* [7]	*Two-Sided Assembly Line Balancing*	Perbandingan antara Algoritma Genetik dan Program Integer	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja.
2	Özcan dan Toklu, 2008 [12]	*Two-Sided Assembly Line Balancing*	*Tabu Search*	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P16 dengan *cycle time* = 22 menghasilkan 4 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun kerja.
3	Wu *et al.*, 2008 [18]	*Two-Sided Assembly Line Balancing*	Algoritma *Branch and Bound*	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun kerja.
4	Baykasoglu dan Dereli, 2006 [3]	*Two-Sided Assembly Line Balancing*	Algoritma *Ant Colony Optimization* (ACO)	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun kerja.
5	Pratama, 2010 [14]	*Two-Sided Assembly Line Balancing*	*Greedy Randomized Adaptive Procedures* (GRASP)	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P16 dengan *cycle time* = 22 menghasilkan 4 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun kerja.
6	Apnena, 2011 [1]	*Two-Sided Assembly Line Balancing*	*Variable Neighbourhood Descent* (VND)	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P16 dengan *cycle time* = 22 menghasilkan 4 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun

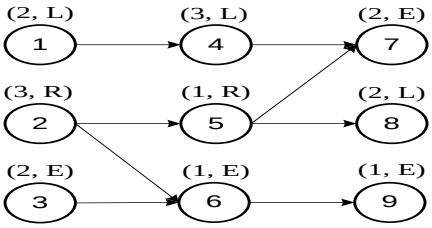
				kerja.
7	Inayah , 2012 [7]	*Two-Sided Assembly Line Balancing*	*Variable Neighbourhood Descent* (VND) *with Fixed Threshold*	P9 dengan *cycle time* = 5 menghasilkan 4 stasiun kerja. P12 dengan *cycle time* = 5 menghasilkan 6 stasiun kerja. P16 dengan *cycle time* = 22 menghasilkan 4 stasiun kerja. P24 dengan *cycle time* = 20 menghasilkan 8 stasiun kerja.

Landasan Teori

Two-sided assembly line balancing (TALB)

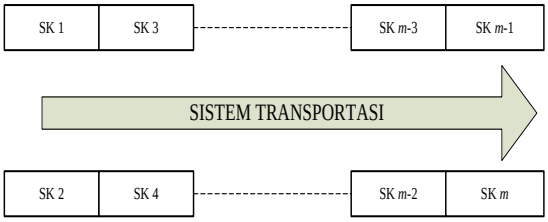
Merupakan pengembangan dari keseimbangan pada lintasan perakitan tunggal (*simple assembly line balancing*, SALB). TALB terdiri dari dua baris lintasan tunggal yang disusun secara paralel, yaitu sisi bagian kiri dan sisi bagian kanan. Elemen kerja pada lintasan perakitan dua sisi dikerjakan untuk kedua sisinya secara bersamaan. Perakitan ini biasanya digunakan untuk perakitan produk yang memiliki jumlah produksi yang besar dan memiliki perbedaan perakitan pada setiap sisinya.

Gambar 1 memperlihatkan bentuk umum *precedence diagram* dari TALB, yang memiliki 5 elemen kerja dengan karakteristik pengerjaannya. Simbol lingkaran merupakan elemen kerja dan garis busur yang menghubungkan dua lingkaran menunjukkan hubungan *precedence*. Setiap elemen kerja ditandai oleh (t_j, d_j) dengan t_j adalah waktu proses elemen kerja ke j dan d_j adalah karakteristik elemen kerja (L = sisi kiri, R = sisi kanan, dan E = salah satu sisi pada lintasan).



Gambar 1. *Precedence Diagram* TALB

Tata letak TALB adalah lintasan perakitan tunggal yang memiliki pasangan SK pada setiap sisinya dengan konveyor sebagai alat trasportasi yang digunakan, m adalah maksimum indeks pada SK dan dapat dilihat pada Gambar 2. Pada setiap sisi terdapat sepasang SK yang secara langsung berhadapan, misalnya SK 1 dan SK 2 merupakan pasangan SK sedangkan SK-1 dan SK 3 merupakan rekan antar SK.



Gambar 2. *Layout* TALB

Operator yang bekerja pada lintasan dua sisi ini ditempatkan pada stasiun kerja paralel yang berpasangan dengan jenis pekerjaan yang berbeda tetapi *item* yang dikerjakannya sama. Setiap pekerjaan dilakukan oleh operator yang berbeda pada setiap stasiun kerjanya, tetapi dapat mempergunakan peralatan yang sama ketika masing-masing pekerjaan tidak menggangu pekerjaan pada sisi yang lainnya. Notasi-notasi yang digunakan dalam *assembly line balancing* adalah:

i indeks untuk stasiun kerja (SK); (i = 1, 2, …, m)
j indeks untuk elemen kerja (EK); (j = 1, 2, …, n)
t_j waktu proses pada elemen kerja ke - j
CT waktu siklus
ST_i akumulasi waktu elemen kerja pada stasiun kerja ke - i
IT_i *idle time* pada stasiun kerja ke-i
IT_{maks} *idle time* dengan nilai terbesar
NY_i *neighboorhood structure* penukar
NX_i *neighboorhood structure* yang ditukar
NI_i *neighboorhood structure* pada proses *insert*
VY_j elemen kerja penukar
VX_j elemen kerja ditukar
VI_j elemen kerja yang yang akan di-*insert*
SI *smoothness index*
EL efisiensi lintasan

Algoritma *Simulated Annealing* (SA)

Algoritma SA diperkenalkan oleh Metropolis *et al*. dalam bidang optimasi algoritma ini beranalogi dengan proses annealing (pendinginan) yang diterapkan dalam pembuatan material yang terdiri dari butir Kristal [10]. Dari sisi ilmu fisika, tujuan sistem ini adalah untuk meminimasi energi potensial. Fluktuasi kinematika

acak menghalangi sistem untuk mencapai energi potensial yang minimum global, sehingga sistem dapat terperangkap dalam sebuah keadaan minimum lokal. Dengan menurunkan temperatur sistem, diharapkan energi dapat dikurangi ke suatu level yang relatif rendah. Semakin lambat laju pendinginan ini, semakin rendah pula energi yang dapat dicapai oleh sistem pada akhirnya.

Untuk menghasilkan perilaku konvergensi dalam algoritma SA, sekumpulan parameter harus didefinisikan terlebih dahulu di awal proses. Cara pendefinisian parameter-parameter ini disebut *cooling schedule*, yang melibatkan:

- nilai awal untuk parameter kontrol temperatur (T_0)
- fungsi/faktor penurunan nilai temperatur (F)
- jumlah iterasi dalam tiap nilai temperatur atau panjang rantai Markov homogen (L)
- nilai akhir untuk temperatur (T_1) atau kriteria terminasi untuk menghentikan eksekusi

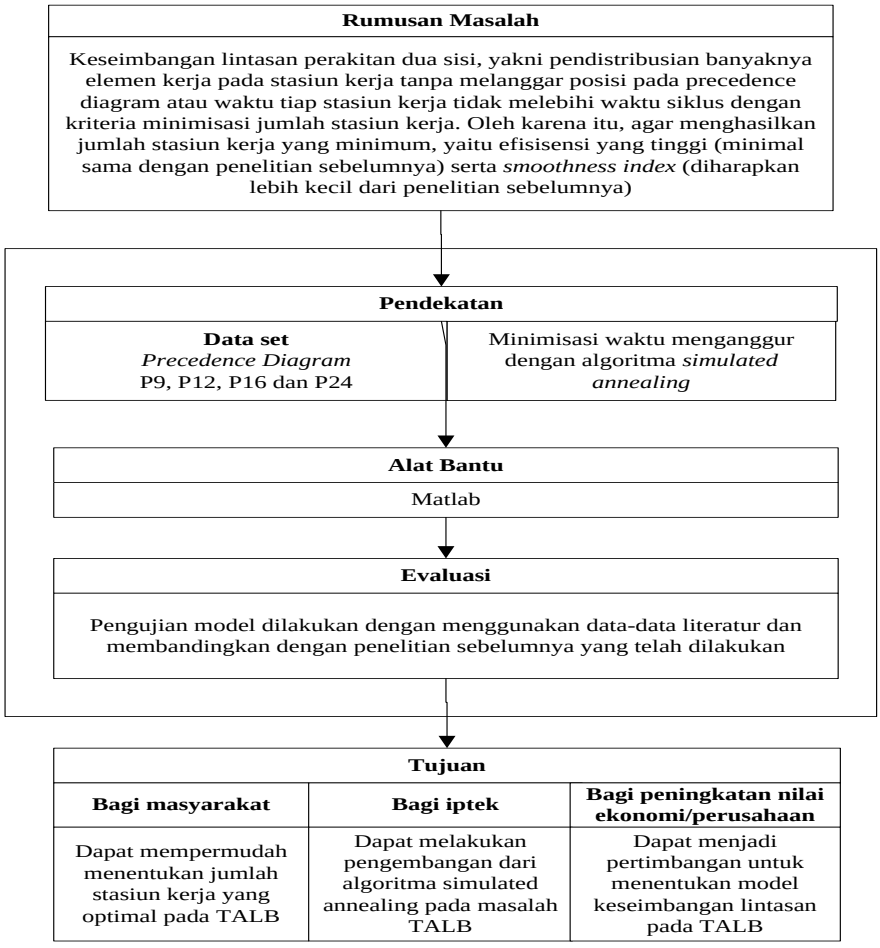
1. Tetapkan skema pendinginan (nilai parameter *annealing*): temperatur awal (T_0) dan temperatur akhir (T_1), faktor penurunan temperatur (F), dan jumlah iterasi dalam tiap nilai temperatur (L).
2. Pilih solusi awal X secara acak atau dengan metode heuristik tertentu.
3. Hitung nilai fungsi objektif: $E = f(X)$.
4. Set nilai parameter kontrol temperatur : $T = T_0$.
5. Ulangi langkah 6 s.d. 8 sebanyak L kali
6. Bangkitkan solusi baru X_{new}
7. Hitung nilai fungsi objektif: $E_{new} = f(X_{new})$.
8. Tetapkan $X = X_{new}$ dan $E = E_{new}$ dengan probabilitas (kriteria Metropolis): min $\{1, exp((f(X) - f(X_{new})) / T)\}$
9. Kurangi nilai parameter kontrol temperatur dengan mengeset : $T = T \times F$
10. Berhenti jika telah dicapai kriteria terminasi; jika tidak kembali ke langkah 5.

Gambar 3. Algoritma umum SA [13]

Kondisi terminasi proses pencarian dalam algoritma SA dapat berupa dicapainya suatu jumlah iterasi tertentu dimana selama itu tidak ada solusi baru yang diterima, atau dicapainya nilai temperatur tertentu yang telah ditetapkan sebelumnya ($T < T_1$). Gambar 3 menunjukkan algoritma SA secara umum.

Kerangka Pemikiran

Gambar 4 merupakan kerangka pemikiran hipotesis dari penelitian ini.



Gambar 4. Kerangka pemikiran hipotesis

III. METODE PENELITIAN

Tahap-tahap dari pengembangan algoritma usulan pada penelitian ini dapat dijelaskan sebagai berikut:

a) Tahap 1 Pembentukan Solusi Inisial

Pada tahap ini dilakukan pembentukan sebuah solusi inisial dengan menggunakan algoritma J-Wagon, cara kerjanya adalah mengurutkan bobot pada setiap elemen kerja berdasarkan jumlah *successor* pada elemen kerja tersebut dari nilai yang terbesar sampai dengan yang terkecil. Pembangkitan solusi inisial dengan menggunakan metode heuristik yaitu, metoda J-Wagon. Berikut langkah-langkahnya:

Langkah 0

Input data j, t_j, CT dan *precedence diagram*

Langkah 1

- Bentuk solusi inisial menggunakan metoda J-Wagon dan simpan sebagai *current solution* (CS). Prosedur dari metoda J-Wagon sebagai berikut:
- Hitung bobot tiap elemen kerja berdasarkan jumlah proses operasi-operasi yang bergantung pada operasi tersebut (*successor*)
- Urutkan bobot elemen kerja dengan mengutamakan jumlah elemen kerja yang terbanyak (*predecessor*), dimana elemen kerja tersebut akan diprioritaskan terlebih dahulu untuk ditempatkan dalam stasiun kerja yang lebih sedikit. Apabila terdapat dua elemen kerja yang memiliki nilai bobot yang sama, maka prioritas akan diberikan kepada elemen kerja yang memiliki waktu pengerjaan lebih besar.
- Tempatkan elemen kerja yang memiliki bobot tertinggi hingga terendah ke setiap stasiun kerja.
- Jika pada setiap stasiun kerja terdapat kelebihan, dalam hal ini waktu stasiun (ST_i) melebihi waktu siklus (CT), tukar atau ganti elemen kerja yang ada dalam stasiun kerja tersebut ke stasiun kerja berikutnya selama tidak menyalahi *precedence diagram*. Hitung nilai *idle time* untuk setiap stasiun kerja.

$$IT = CT - ST_i \quad \text{.....(1)}$$

- Ulangi Langkah 1.3 dan 1.4 sampai seluruh elemen pekerjaan sudah ditempatkan ke dalam stasiun kerja.

Langkah 2

Buatlah *layout* dari hasil perhitungan menggunakan metoda J-Wagon kemudian hitung efisiensi lintasan (EL), dan *smoothness index* (SI) dan di set menjadi $E = f(X)$.

$$EL = \frac{\sum_{i=1}^{m} ST_i}{m \times CT} \quad \text{.....(2)}$$

$$SI\sqrt{\sum_{i=1}^{m}(ST_{maks} - ST_i)^2} \quad \text{.....(3)}$$

b) Tahap 2 *Local Search*

Tahap ini merupakan tahap pencarian solusi yang lebih baik pada *neighborhood* yang terbentuk dari tahap solusi inisial dengan melakukan modifikasi atau perubahan struktur *neighborhood*. Pada penelitian ini perubahan struktur *neighborhood* dibagi menjadi 2 proses, yaitu:

- *Exchange q*(*Intra dan extra exchange*)
- *Insert*

Tahap *local search* menggunakan metode metaheuristik, yaitu algoritma *simulated annealing*. Berikut langkah-langkahnya:

Langkah 3

Input data j, t_j, CT, ST_i, EL, SI dari solusi inisial dan T_0, T_1, F sebagai parameter *annealing*.

Langkah 4

Set $T = T_0 = 1000$, $T_1 = 500$ dan $F = 0{,}5$

Langkah 5

Set $L = 1$ dan $L_{maks} = m$

Langkah 6

Tentukan *idle time* (IT) setiap stasiun kerja dengan persamaan

$$IT_i = \{CT - ST_i\} \quad \text{.....(4)}$$

Langkah 7

Pilih nilai IT yang terbesar (IT_{maks}) dengan menggunakan persamaan

$$IT_{maks} = {}^{maks}_{\forall_i}\{IT_i\} \quad \text{.....(5)}$$

Set NX_i dengan i yang terpilih berdasarkan IT terbesar. Jika terdapat nilai IT_i yang sama maka pilih stasiun kerja terbesar, lanjutkan ke Langkah 8

Langkah 8

Apakah $|j \in NX_i| = 1$?

Jika ya, Set VX_j. Lanjutkan ke Langkah 10

Jika tidak, lanjutkan ke Langkah 9

Langkah 9

Pilih $VX_j, j \in NX_i$ menggunakan persamaan

$$VX_j = arg^{\,maks}_{j \in NX_i}\{t_j\} \quad (6)$$

Set VX_j dengan j yang terpilih berdasarkan t_j terbesar. Jika terdapat t_j yang sama maka pilih secara stasiun kerja terbesar, lanjutkan ke Lanjutkan ke Langkah 10

Langkah 10

Apakah terdapat posisi j yang memungkinkan/sama dengan VX_j pada NY_i untuk dilakukan proses *exchange*? Dengan tidak melanggar *precedence constraint* atau $ST_i > CT$?

Jika ya, Lanjutkan ke Langkah 10.1

Jika tidak, lanjutkan ke Langkah 11

Langkah 10.1

Apakah $VY_j > 1$?

Jika ya, pilih j dengan nilai SI terkecil. Jika terdapat SI yang sama, maka pilih secara random, kemudian set $j = VY_j$ dan *set* $i = NYi$. Lanjutkan ke Langkah 10.2

Jika tidak, set $j = VY_j$ dan *set* $i = NYi$. Lanjutkan ke Langkah 10.2

Langkah 10.2

Lakukan proses *exchange* antara VX_j dan VY_j dan lanjutkan ke Langkah 12

Langkah 11

Apakah masih terdapat $j \in NX_i$ yang belum ditukar?

Jika ya, kembali ke Langkah 9

Jika tidak, set $L = L + 1$ dan lanjutkan ke Langkah 14

Langkah 12

Apakah terdapat NX_i dengan $j = 1$?

Jika ya, lanjutkan ke Langkah 13

Jika tidak, set $L = L + 1$ dan lanjutkan ke Langkah 14

Langkah 13

Apakah j tersebut dapat dilakukan proses *insert* pada NX_i yang lain?

Jika ya, lakukan *insert* j tersebut pada NX_i yang dituju dan lanjutkan ke Langkah 16

Jika tidak, set $L = L + 1$ dan lanjutkan ke Langkah 14

Langkah 14

Apakah $L > L_{maks}$?

Jika ya, lanjutkan ke Langkah 15

Jika tidak, kembali ke Langkah 6

Langkah 15

Apakah terdapat j pada NX_i yang dapat dilakukan *intra-exchange* tanpa melanggar *precedence constraint*?

Jika ya, lakukan proses *intra-exchange* pada semua j yang memungkinkan pada setiap NX_i dan kembali ke Langkah 12

Jika tidak, lanjutkan ke Langkah 16

Langkah 16

Hitung $E_{new} = f(X_{new})$ dan lanjutkan ke Langkah 17

Langkah 17

Apakah $f(X_{new}) > f(X)$?

Jika ya, hitung

$$P\{\text{Terima } X_{new}\} = exp\frac{f(X)-f(Xnew)}{T} \quad \text{.....(7)}$$

Kemudian set $L = L + 1$ dan lanjutkan ke Langkah 14

Jika tidak, $P\{\text{Terima } X_{new}\} = 1$. Set $E_{new} = E$, lanjutkan ke Langkah 18

Langkah 18

Kurangi nilai temperature dengan set $T = T$ x F dan lanjutkan ke Langkah 19

Langkah 19

Apakah $T \leq T_1$?

Jika ya, berhenti dan tampilkan *output* konfigurasi terbaik

Jika tidak, kembali ke Langkah 4

Langkah 20

Lakukan proses evaluasi dari hasil konfigurasi terbaik

Langkah 21

Lakukan analisis dari hasil konfigurasi terbaik

IV. HASIL DAN PEMBAHASAN

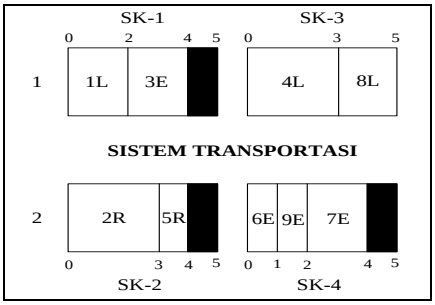
P*recedence Diagram* dengan 9 Elemen Kerja (P9)

Hasil eksperimen dari P9 dapat dilihat pada Tabel 2 dan layout hasil eksperimen dapat dilihat pada Gambar 5.

Tabel 2. Hasil eksperimen dari P9

Stasiun Kerja (i)	Elemen Kerja (j)	Waktu Proses (t_j)	ST_i	CT	IT
SK-1	1	2	4	5	1
	3	2			
SK-3	4	3	5	5	0
	8	2			
SK-5	7	2	2	5	3

Stasiun Kerja (i)	Elemen Kerja (j)	Waktu Proses (t_j)	ST_i	CT	IT
SK-2	2	3	4	5	1
	5	1			
SK-3	6	1	2	5	3
	9	1			



Gambar 5. *Layout* hasil eksperimen dari P9

Perhitungan Efisiensi Lintasan (*EL*) dan Smoothness Index (*SI*)

$EL = \frac{\sum_{i=1}^{m} ST_i}{m \times CT} = \frac{(4+4+5+2+2)}{5 \times 22}$ = 85%

$SI = \sqrt{\sum_{i=1}^{m} (ST_{maks} - ST_i)^2} = \sqrt{(1^2 + 1^2 + 0^2 + 1^2)}$ = 1,73

Dari hasil eksperimen didapat nilai efisiensi lintasan yaitu 85% yang berarti total waktu stasiun kerja berbanding dengan waktu siklus dikali dengan jumlah stasiun kerja yang terbentuk. Semakin sedikit jumlah stasiun kerja yang terbentuk maka akan semakin besar nilai efisiensi lintasan yang didapat. Serta dari hasil eksperimen didapat juga nilai *smoothness index* yaitu 1,73 yang berarti kelancaran pada lintasan perakitan yang terbentuk sangat baik, karena jika semakin kecil nilai *SI* yang terbentuk (mendekati nol) artinya semakin lancar pula proses perakitan pada lintasan tersebut. Hubungan yang terbentuk antara *EL* dan *SI* yaitu berbanding terbalik. Semakin besar nilai *EL* maka akan semakin kecil nilai SI yang didapat.

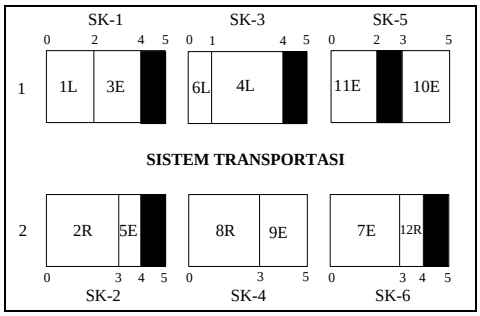
P*recedence Diagram* dengan 12 Elemen Kerja (P12)

Hasil eksperimen dari P12 dapat dilihat pada Tabel 3 dan layout hasil eksperimen dapat dilihat pada Gambar 6.

Tabel 3. Hasil eksperimen dari P12

Stasiun Kerja (*i*)	Elemen Kerja (*j*)	Waktu Proses (t_j)	ST_i	*CT*	*IT*
SK-1	1	2	4	5	1
	3	2			
SK-3	6	1	4	5	1
	4	3			
SK-5	11	2	4	5	1

Stasiun Kerja (*i*)	Elemen Kerja (*j*)	Waktu Proses (t_j)	ST_i	*CT*	*IT*
SK-2	2	3	4	5	1
	5	1			
SK-4	8	3	5	5	0
	9	2			
SK-6	7	3	4	5	1



Gambar 6. *Layout* hasil eksperimen dari P12

Perhitungan Efisiensi Lintasan (*EL*) dan Smoothness Index (*SI*)

$EL = \frac{\sum_{i=1}^{m} ST_i}{m \times CT} = \frac{(3+5+4+5+3)}{5 \times 5}$ = 83,333 %

$SI = \sqrt{\sum_{i=1}^{m} (ST_{maks} - ST_i)^2} =$

$\sqrt{(1^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2)}$ = 2,236

Dari hasil eksperimen didapat nilai efisiensi lintasan yaitu 83,333% yang berarti total waktu staisun kerja berbanding dengan waktu siklus dikali dengan jumlah stasiun kerja yang terbentuk. Semakin sedikit jumlah stasiun kerja yang terbentuk maka akan semakin besar nilai efisiensi lintasan yang didapat. Serta dari hasil eksperimen didapat juga nilai *smoothness index* yaitu 2,236 yang berarti kelancaran pada lintasan perakitan yang terbentuk baik, karena jika semakin kecil nilai *SI* yang terbentuk (mendekati nol) artinya semakin lancar pula proses perakitan

pada lintasan tersebut. Hubungan yang terbentuk antara *EL* dan *SI* yaitu berbanding terbalik. Semakin besar nilai *EL* maka akan semakin kecil nilai SI yang didapat.

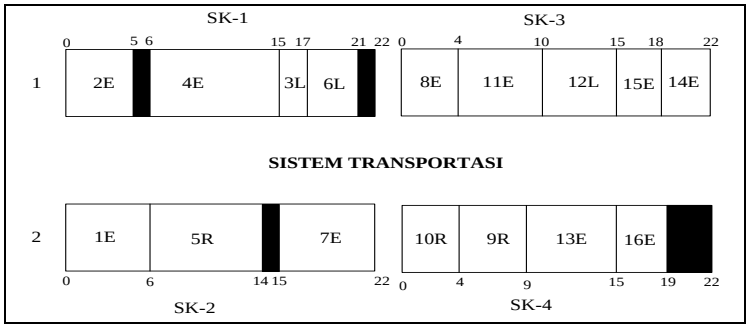
P*recedence Diagram* dengan 16 Elemen Kerja (P16)

Hasil eksperimen dari P16 dapat dilihat pada Tabel 4 dan layout hasil eksperimen dapat dilihat pada Gambar 7.

Tabel 4. Hasil eksperimen dari P16

Stasiun Kerja (*i*)	Elemen Kerja (*j*)	Waktu Proses (t_j)	ST_i	*CT*	IT_i
SK-1	2	5	20	22	2
	4	9			
	3	2			
	6	4			
SK-3	8	4	18	22	4
	11	6			
	12	5			
	15	3			
	14	4			

Stasiun Kerja (*i*)	Elemen Kerja (*j*)	Waktu Proses (t_j)	ST_i	*CT*	IT_i
SK-2	1	6	21	22	1
	5	8			
	7	7			
SK-4	10	4	19	22	3
	9	5			
	13	6			
	16	4			



Gambar 7. *Layout* hasil eksperimen dari P16

Perhitungan Efisiensi Lintasan (*EL*) dan Smoothness Index (*SI*)

$EL = \frac{\sum_{i=1}^{m} ST_i}{m \times CT} = \frac{(21+20+22+19)}{4 \times 22}$ = 93,182 %

$SI = \sqrt{\sum_{i=1}^{m} (ST_{maks} - ST_i)^2} = \sqrt{(1^2 + 2^2 + 0^2 + 3^2)}$ =3,742

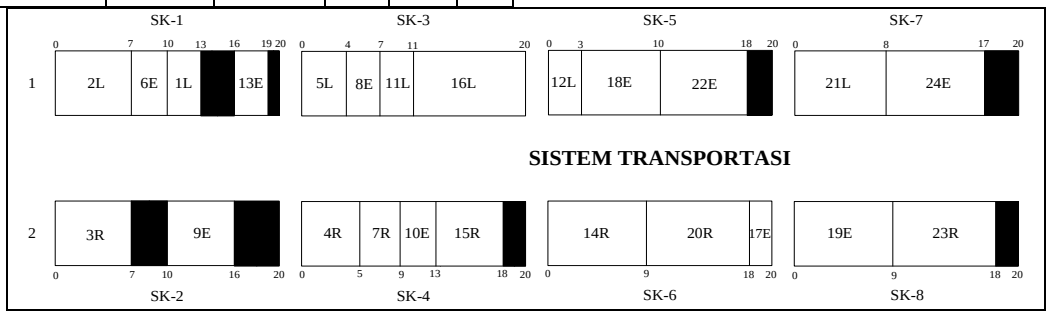
Dari hasil eksperimen didapat nilai efisiensi lintasan yaitu 93,182% yang berarti total waktu stasiun kerja berbanding dengan waktu siklus dikali dengan jumlah stasiun kerja yang terbentuk. Semakin sedikit jumlah stasiun kerja yang terbentuk maka akan semakin besar nilai efisiensi lintasan yang didapat. Serta dari hasil eksperimen didapat juga nilai *smoothness index* yaitu 3,742 yang berarti kelancaran pada lintasan perakitan yang terbentuk sangat baik, karena jika semakin kecil nilai *SI* yang terbentuk (mendekati nol) artinya semakin lancar pula proses perakitan pada lintasan tersebut. Hubungan yang terbentuk antara *EL* dan *SI* yaitu berbanding terbalik. Semakin besar nilai *EL* maka akan semakin kecil nilai SI yang didapat.

P*recedence Diagram* dengan 24 Elemen Kerja (P24)

Hasil eksperimen dari P24 dapat dilihat pada Tabel 5 dan layout hasil eksperimen dapat dilihat pada Gambar 8.

Tabel 5. Hasil eksperimen dari P24

Stasiun Kerja (*i*)	Elemen Kerja (*j*)	Waktu Proses (t_j)	ST_i	*CT*	*IT*
SK-1	2	7	16	20	4
	6	3			
	1	3			
	13	3			
SK-3	5	4	20	20	0
	8	3			
	11	4			
	16	9			
SK-5	12	3	18	20	2
	18	7			
	22	8			
SK-7	21	8	17	20	3
	24	9			
SK-2	3	7	13	20	7
	9	6			
SK-4	4	5	18	20	2
	7	4			
	10	4			
	15	5			
SK-6	14	9	20	20	0
	20	9			
	17	2			
SK-8	19	9	17	20	3
	23	9			



Gambar 8. *Layout* hasil eksperimen dari P24

Perhitungan Efisiensi Lintasan (*EL*) dan Smoothness Index (*SI*)

$$EL = \frac{\sum_{i=1}^{m} ST_i}{m \times CT} = \frac{(16+13+20+18+18+20+17+18)}{8 \times 20} = 87{,}5\ \%$$

$$SI = \sqrt{\sum_{i=1}^{m} (ST_{maks} - ST_i)^2} = \sqrt{(4^2 + 7^2 + 0^2 + 2^2 + 2^2 + 0^2 + 3^2 + 2^2)} = 9{,}274$$

Dari hasil eksperimen didapat nilai efisiensi lintasan yaitu 87,5% yang berarti total waktu staisun kerja berbanding dengan waktu siklus dikali dengan jumlah stasiun kerja yang terbentuk. Semakin sedikit jumlah stasiun kerja yang terbentuk maka akan semakin besar nilai efisiensi lintasan yang didapat. Serta dari hasil eksperimen didapat juga nilai *smoothness index* yaitu 9,274 yang berarti kelancaran pada lintasan perakitan yang terbentuk cukup baik, karena jika semakin kecil nilai *SI* yang terbentuk (mendekati nol) artinya semakin lancar pula proses perakitan pada lintasan tersebut. Hubungan yang terbentuk antara *EL* dan *SI* yaitu berbanding terbalik. Semakin besar nilai *EL* maka akan semakin kecil nilai SI yang didapat.

V. KESIMPULAN DAN SARAN

Kesimpulan

Pada penelitian ini jumlah stasiun kerja yang terbentuk dilakukan hanya pada sekali putaran (dengan T_0 = 1000, T_1 = 500 dan F = 0,5) sudah mendapatkan jumlah stasiun kerja yang sama dengan penelitian-penelitian yang dilakukan sebelumnya, yaitu 4 stasiun kerja pada jumlah elemen kerja 9 dan 16, 6 stasiun kerja pada jumlah elemen kerja 12 dan 8 stasiun kerja dengan jumlah elemen kerja 24. Pada efisiensi lintasan yang didapat dilakukan hanya pada sekali putaran (dengan T_0 = 1000, T_1 = 500 dan F = 0,5) sudah mendapatkan efisiensi lintasan yang sama dengan penelitian-penelitian yang dilakukan sebelumnya, yaitu 85% pada jumlah elemen kerja 9; 83,333% pada jumlah elemen kerja 12; 93,18% pada jumlah stasiun kerja 16 dan 87,5% pada jumlah stasiun kerja. Pada *smoothness index* yang didapat dilakukan hanya pada sekali putaran (dengan T_0 = 1000, T_1 = 500 dan F = 0,5) sudah mendapatkan *smoothness index* yang sama dengan penelitian-penelitian yang dilakukan sebelumnya, yaitu 1,7 pada jumlah elemen kerja 9; 2,6 pada jumlah elemen kerja 12 dan 3,7 pada jumlah stasiun

kerja 16. Namun, pada jumlah elemen kerja 24 didapat *smoothness index* 9,3 yang berarti lebih baik dari penelitian dengan menggunakan algoritma GRASP tetapi tidak lebih baik dari penelitian VND.

Saran

Saran untuk penelitian selanjutnya dapat mengembangkan algoritma metaheuristik lainnya untuk menyelesaikan masalah keseimbangan lintasan perakitan *two-sided* dengan minimisasi jumlah stasiun kerja berdasarkan *SI* yang didapat.

DAFTAR PUSTAKA

Apnena, R. D.,(2011). *Model Keseimbangan Lintasan Perakitan Two-Sided Menggunakan Algoritma Variable Neighbourhood Descent dengan Kriteria Minimisasi Jumlah Stasiun Kerja*. Skripsi : *Institut Teknologi Nasional*.

Bartholdi, J. J.,(1993). *Balancing two-sided assembly lines: a case study*. International Journal Prodction Research 31(10) : 2447–2461.

Baykasoglu, A., dan Dereli, T.(2006). *Two-Sided Assembly Line Balancing Using An Ant-Colony-Based Heuristic*. International Journal of Advanced Manufacturing Technology.

Baykasoglu, A., dan Dereli, T.(2006). *Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems*. *J Intell Manuf,* 17:217–232.

Glover, F dan Laguna, M.,(1997). *Principles of Tabu Search.* Kluwer Academic Publishers. Book on metaheuristics and in particular tabu search.

Gutjhar, A. L., dan Nemhauser, G. L.,(1964). *An Algorithm for The Line Balancing Problem*. *Management Science*, 11 :308–315.

Inayah,(2012). *Algoritma Variable Neighborhood Descent With Fixed Threshold untuk Keseimbangan Lintasan Perakitan Two Sided dengan Kriteria Minimisasi Jumlah Stasiun Kerja.* Jurnal Online Institut Teknologi Nasional, Vol. 1, No. 3, Maret 2013.

Kim, Y. K., Kim, Y., dan Kim, Y.J., (2000). *Two-sided assembly line balancing: a genetic algorithm approach*. Production Planning and Control 11 (1),44–53.

Kim, Y. K., Song, W. S., dan Kim, J. H.,(2009). *A mathematical model and a genetic algorithm for two-sided assembly line balancing*. Computers and Operations Research, 36(3), 853–865.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller,(1953). *Equation of State Calculations by Fast Computing Machines,* J. Chem. Phys., Vol. 21, No. 6, pp. 1087 - 1092.

Osman, I. H., Kelly, J. P.,(1996). *Meta-heuristics: An Overview.* Dalam Osman, I. H., Kelly, J. P., editor. Meta-heuristics: Theory and Application. Kluwer Academic Publishers, Massachusetts, 1-21.

Ozcan, U., dan Toklu, B.,(2008). *A Tabu Search Algorithm for Two-Sided Assembly Line Balancing*. International Journal of Advanced Manufacturing Technology, 1753-5.

Panggabean, H. P.,(2004). *Algoritma Simulated Annealing untuk Pembentukan Sel Mesin dengan Dua Tipe Fungsi Objektif dan Dua Cara Pembatasan Sel*. Jurnal Teknik Industri , Vol. 6, No. 1, 10 – 24.

Pratama, T. G.,(2010). *Model Keseimbangan Lintasan Perakitan Two Sided menggunakan Algoritma Greedy Randomized Adaptive Procedures dengan Kriteria Minimisasi Jumlah Stasiun Kerja*. *Institut Teknologi Nasional*.

Reeves, C. R., Beasley, J. E.,(1993). *Introduction. In Reeves, C. R., editor, Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, Oxford, 1-19.

Scholl, A.,(1999). *Balancing and Sequencing of Assembly Lines*. Second Edition. Physica-Verlag Heidelberg New York.

Suresh, G., Sahu, S.,(1994). *Stochastic Assembly Line Balancing Using Simulated Annealing*. Int. J. Prod. Res., Vol. 32, No. 8, 1801-1810.

Wu, E.-F., Jin, Y., Bao, J.-S., dan Hu, X.-F.,(2008). A branch-and-bound algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, 39, 1009 1015.