

Algoritma Variable Neighborhood Descent with Fixed Threshold* untuk Keseimbangan Lintasan Perakitan *Mixed Model* Dengan Kriteria Minimisasi Jumlah Stasiun Kerja

RENITA YUNIAR, ARIF IMRAN, ALEX SALEH

Jurusan Teknik Industri
Institut Teknologi Nasional (Itenas) Bandung

Email:renitayuniar@gmail.com

ABSTRAK

Keseimbangan lintasan perakitan mixed model adalah lintasan produksi dimana berbagai macam model produk yang dirakit memiliki karakteristik serupa. Produk yang dihasilkan dalam MMAL biasanya memiliki perbedaan dalam jumlah produksi, isi kerja, dan waktu perakitan tergantung pada model. Tujuan dari model tersebut adalah menugaskan elemen kerja ke dalam beberapa stasiun kerja dengan memperhatikan hubungan precedence. Metode yang digunakan pada penelitian ini adalah algoritma variable neighborhood descent with fixed threshold. Variable Neighborhood Descent (VND) with fixed threshold adalah algoritma yang digunakan untuk menyelesaikan permasalahan optimasi kombinatorial dengan cara melakukan perubahan pada struktur neighborhood. Setiap perubahan struktur neighborhood yang menghasilkan nilai yang lebih baik dari sebelumnya atau lebih kecil tetapi masih dalam batas ambang threshold maka diterima dan digunakan untuk iterasi selanjutnya. Perubahan struktur neighborhood ini dilakukan dengan cara exchange dan insert.

Kata kunci: *Assembly Line Balancing, Variable Neighborhood Descent, Threshold Accepting, Mixed Model*

ABSTRACT

Mixed model assembly line (MMAL) is a line production of a variety of models that are assembled product with similar characteristics. Products who produced in MMAL usually have a difference in the amount of production, the content of work, and assembly time depends on the model. The goal is to assign work elements into multiple work stations with respect to precedence relationships. The method used in this research was variable neighborhood descent algorithm

* Makalah ini merupakan ringkasan dari Tugas Akhir yang disusun oleh penulis pertama dengan pembimbingan penulis kedua dan ketiga. Makalah ini merupakan draft awal dan akan disempurnakan oleh para penulis untuk disajikan pada seminar nasional dan/atau jurnal nasional.

with fixed threshold. Variable neighborhood descent (VND) algorithm with fixed threshold is used to solve combinatorial optimization problems by making changes to the structure of the neighborhood. Any changes in neighborhood structure that produces a better value than before or smaller but still within the acceptable threshold and the threshold used for the next iteration. Changes in the structure of neighborhood are done by way of exchange and insert.

Keyword: *Assembly Line Balancing, Variable Neighborhood Descent, Threshold Accepting, Mixed Model*

1. PENDAHULUAN

Keseimbangan lintasan perakitan (*Assembly Line Balancing*, ALB) adalah proses penempatan sejumlah pekerjaan ke dalam stasiun-stasiun kerja yang saling berkaitan dalam satu lintasan produksi, sehingga setiap stasiun kerja memiliki waktu yang tidak melebihi waktu siklus dari stasiun kerja tersebut. Permasalahan yang diamati dalam model lintasan perakitan disebut *Simple Assembly Line Balancing* (SALB). Solusi untuk pendekatan ini memiliki dua variasi yaitu, *Simple Assembly Line Balancing Problem I* (SALBP-I) dan *Simple Assembly Line Balancing Problem* (SALBP II) (Scholl, 1999). Permasalahan pada SALBP-I merupakan permasalahan untuk meminimasi jumlah stasiun kerja dengan sasaran waktu siklus yang dapat terpenuhi sedangkan SALBP-II merupakan permasalahan untuk meminimasi waktu siklus, dimana jumlah stasiun kerjanya telah ditetapkan. Pada kondisi nyata, penggunaan lintasan perakitan tunggal (SALB) sangat tidak efisien untuk perusahaan yang memproduksi secara massal karena jumlah permintaan yang banyak dipengaruhi oleh pengaruh pasar. Saat ini persaingan pasar sangatlah kompetitif dan intensif sehingga produsen mencoba untuk menarik setiap pelanggan dengan lebih membuat variasi dari produk yang ada. Maka dari itu muncul permasalahan model lintasan perakitan *mixed model* yang disebut *Mixed Model Assembly Line Balancing* (MALB).

Permasalahan MALB ini telah diamati selama 4 dekade. Penelitian terbaru yang dilakukan untuk menyelesaikan MALB adalah menggunakan *Variable Neighborhood Descent* (VND) oleh Yohana (2011). Algoritma VND adalah sebuah metode yang digunakan untuk menyelesaikan permasalahan optimisasi kombinatorial VND dengan melakukan perubahan *neighborhood* secara deterministik. Kekurangan pada penelitian ini adalah VND tidak mempertimbangan perubahan struktur *neighborhood* yang menghasilkan solusi yang lebih jelek dari solusi sebelumnya. Hanya perubahan struktur *neighborhood* yang menghasilkan solusi lebih baik saja yang digunakan untuk proses iterasi selanjutnya. Memperhatikan hal tersebut, maka perlu dikembangkan suatu algoritma yang dapat menerima solusi yang lebih jelek terlebih dahulu, dengan harapan akan membuka peluang untuk menghasilkan solusi yang lebih baik. Maka dilakukan penelitian mengenai algoritma *Variable Neighborhood Descent* (VND) *with fixed threshold. Threshold Accepting* (TA) ini adalah sebuah metode metaheuristik yang diperkenalkan oleh Dueck dan Scheuer (1990) yang merupakan bagian dari deterministik *threshold*. Dalam metode ini semua solusi yang terbaik diterima dan lebih rendah yaitu yang memiliki nilai lebih kecil tetapi masih dalam batas ambang atau sama dengan batas ambang juga diterima.

Tujuan dari penelitian ini menghasilkan algoritma *Variable Neighborhood Descent* (VND) *with fixed threshold* untuk menyelesaikan permasalahan dalam lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan cara maksimasi efisiensi lintasan. Jika

efisiensi lintasan sudah maksimum (tidak adanya waktu menganggur disetiap stasiun kerja) menandakan stasiun kerja yang dihasilkan sudah sangat minimum.

Batasan-batasan masalah yang dilakukan pada penelitian ini adalah:

1. Jumlah operator yang diperlukan untuk merakit produk tidak dibahas.
2. Waktu *set up* operasi tidak dibahas.
3. Sistem yang dibahas hanya lintasan perakitan *mixed model*.

Asumsi-asumsi yang digunakan dalam penelitian ini adalah:

1. Waktu proses setiap variasi konstan.
2. Tidak diperbolehkan terdapat *buffer* antar stasiun.
3. Beban kerja di setiap stasiun kerja tidak diperhitungkan.

2. METODOLOGI PENELITIAN

Metodologi penelitian merupakan penjabaran langkah-langkah yang digunakan dalam pengembangan algoritma secara terperinci sehingga mendapatkan hasil yang terarah dan terstruktur dalam pemecahan masalah, sehingga sesuai dengan tujuan penelitian. Adapun tahapan dalam penelitian ini adalah sebagai berikut.

2.1 Studi Literatur

Pada tahap ini ditinjau beberapa literatur yang digunakan pada penelitian yaitu *Single Line Assembly Line Balancing (SALBP)*, *Mixed Model Assembly Line Balancing (MALB)*, algoritma *Variable Neighborhood Descent (VND)* dan *Threshold Accepting (TA)* yang akan digunakan untuk menyelesaikan permasalahan MALBP dengan kriteria minimisasi jumlah stasiun kerja diantaranya adalah:

1. Scholl dan Becker (2006) mendefinisikan permasalahan *Simple Line Assembly Line Balancing Problem (SALBP)* menjadi tiga, yaitu SALBP-I, SALBP-II, SALBP-E.
2. Gokcen & Erel (1997) memecahkan permasalahan MALB menggunakan metode optimasi dengan menggunakan *Binary integer programming*.
3. Hansen dan Maldenovic (1999) yang mengembangkan algoritma *Variable Neighborhood Descent (VND)*. Algoritma ini menggunakan set *neighborhood* dalam melakukan proses *local search* untuk mendapatkan solusi optimum dengan melakukan perubahan pada *neighborhood*.
4. Yohana (2011) menggunakan algoritma *Variable Neighborhood Descent (VND)* untuk menyelesaikan MALBP.
5. Dueck and Scheuer (1990) melakukan penelitian pertama kali tentang *threshold accepting* untuk menyelesaikan permasalahan optimasi kombinatorial.

2.2 Tahap Identifikasi Posisi Penelitian

Berdasarkan literatur, saat ini belum dilakukan penelitian mengenai pengembangan algoritma *variable neighborhood descent with fixed threshold* untuk menyelesaikan permasalahan *mixed model assembly line balancing*. Maka perlu dilakukan penelitian mengenai pengembangan algoritma tersebut. Posisi penelitian ini terhadap penelitian-penelitian lain dapat dilihat pada Gambar 1.

2.3 Tahap Pengembangan Algoritma

Pengembangan algoritma yang dilakukan dalam penelitian ini adalah membuat rancangan lintasan perakitan *mixed-model* menggunakan algoritma VND dengan kriteria minimisasi

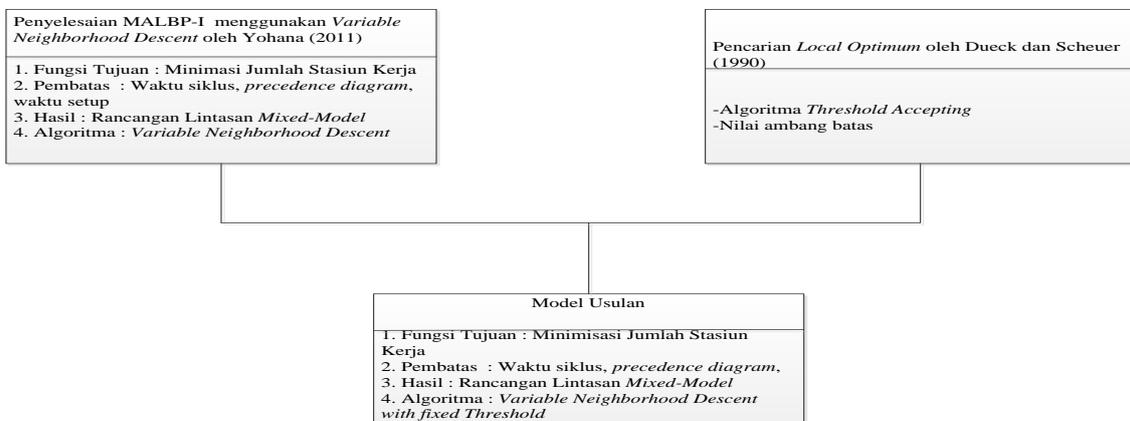
jumlah stasiun kerja. Penelitian ini merupakan pengembangan algoritma dari penelitian-penelitian yang telah dilakukan sebelumnya. Penelitian yang menjadi dasar dari penelitian ini diantaranya adalah:

1. Gocken & Erel (1997) yang membahas penyelesaian permasalahan *mixed model assembly line balancing* dengan menggunakan *Binary Integer Formulation* untuk meminimisasi jumlah stasiun kerja.
2. Hansen dan Maldenovic (1999) yang mengembangkan algoritma *Variable Neighborhood Descent* (VND).
3. Dueck dan Scheuer (1990) mengembangkan sebuah metode metaheuristik yang merupakan modifikasi dari simulasi *annealing* dimana menggunakan fungsi probabilistik dalam menerima *nonimproving* solusi.

Minimasi Jumlah Stasiun Kerja							
Keseimbangan Lintasan Perakitan	Lintasan	Optimasi	Heuristik	Metaheuristik			
				VND	GRAPS	Guided GRAPS	VND with fixed threshold
	SALBP	Elsayed dan Boucher (1994)	Ponnambalam <i>et al</i> (1999)	Rahadian (2010)	Andres <i>et al</i> (2008)		
MALBP	Gocken dan Erel (1997)	Thomopoulos (1970)	Yohana (2011)	Putrianti (2011)	Rahayu (2011)		Penelitian

Gambar 1. Peta Posisi Penelitian

Skema pengembangan model dapat dilihat pada Gambar 2.



Gambar 2. Skema Pengembangan Model

2.4 Pengujian Algoritma dan Analisis

Pengujian algoritma ini dilakukan untuk mengetahui performansi dari algoritma usulan dengan menggunakan tiga skenario. Skenario yang digunakan dalam pengujian algoritma usulan pada penelitian ini adalah:

1. Skenario 1 digunakan untuk mengetahui cara kerja dari model usulan yang telah dibuat. Skenario 1 ini menggunakan set data dari penelitian Su dan Lu (2007) yang terdapat dalam Tugas Akhir Yohana (2011), terdiri dari 3 variasi produk dengan jumlah elemen kerja (n) = 17.

2. Skenario 2 digunakan untuk menguji keandalan dari model usulan dengan menggunakan set data penelitian Thomopoulos (1970) yang terdiri dari 3 variasi produk dengan jumlah elemen kerja (n) = 19 .
3. Skenario 3 Bedworth dan Bailey (1982) yang terdiri dari 2 variasi produk yaitu variasi 1 dan variasi 2 dengan jumlah elemen kerja (n) = 11 dan waktu siklus untuk masing-masing model (CT) = 10.

Untuk dapat menganalisis performansi dari algoritma usulan ini, maka dilakukan perbandingan dengan beberapa penelitian yang telah dipublikasikan.

2.5 Kesimpulan dan Saran

Bagian ini berisikan mengenai kesimpulan yang diperoleh dari hasil pengembangan algoritma dan saran-saran yang ditujukan untuk pengembangan selanjutnya.

3. PENGEMBANGAN ALGORITMA

Penelitian ini bertujuan untuk menghasilkan sebuah lintasan perakitan *mixed-model* menggunakan algoritma VND *with fixed threshold* dengan kriteria maksimasi efisiensi lintasan. Penelitian-penelitian terdahulu dijadikan algoritma dasar dalam penelitian ini. Dalam pengembangan algoritma pada penelitian ini, notasi-notasi yang digunakan adalah sebagai berikut:

- i = indeks untuk stasiun kerja (SK); ($i = 1, 2, \dots, m$)
 - j = indeks untuk elemen kerja; ($j = 1, 2, \dots, n$)
 - z = indeks untuk variasi; ($z = 1, 2, \dots, z$)
 - t_j = waktu proses pada elemen kerja ke- j
 - CT = waktu siklus
 - CT_z = waktu siklus di setiap variasi ke- z
 - ST_i = akumulasi waktu elemen kerja pada stasiun kerja ke- i
 - ST_{iz} = akumulasi waktu elemen kerja pada stasiun kerja ke- i di variasi z
 - EL = efisiensi lintasan
 - SI = *smoothness index*
 - IT_i = *idle time* pada stasiun kerja ke- i
 - IT_{max} = *idle time* dengan nilai terbesar
 - NI_i = *neighborhood structure* yang di *insert* pada stasiun kerja ke- i
 - NX_i = *neighborhood structure* yang ditukar pada stasiun kerja ke- i
 - NY_i = *neighborhood structure* penukar pada stasiun kerja ke- i
 - VX_j = elemen kerja yang ditukar
 - VY_j = elemen kerja penukar
 - e = indeks iterasi untuk proses *insert* ($e = 1, 2, \dots, e_{maks}$)
 - EL_{Th} = efisiensi lintasan terbaik * % Nilai T_h
- Nilai T_h = batas ambang dari Nilai T_h yang diizinkan dalam penerimaan solusi yang dilakukan pada tahap *local search*.

Berikut ini merupakan tahap-tahap dalam menyelesaikan keseimbangan lintasan perakitan *mixed model* menggunakan algoritma *variable neighborhood descent with fixed threshold*:

Tahap 1. Initial Solution

Langkah 0

Inputkan j , t_j , CT , dan *precedence diagram*.

Langkah 1

Lakukan proses penggabungan *precedence diagram* dengan menggunakan *precedence matrices* sehingga didapat *precedence diagram* gabungan.

Langkah 2

Buatlah solusi inisial dari permasalahan yang ada dengan menggunakan metoda J-Wagon dan simpanlah solusi tersebut sebagai *current solution*.

Berikut langkah-langkah initial solution menggunakan metode J-Wagon:

- 2.1. Buat *precedence diagram* dari proses yang bersangkutan.
- 2.2. Hitung bobot tiap elemen kerja berdasarkan jumlah *successor* pada operasi tersebut.
- 2.3. Urutkan bobot elemen kerja dengan mengutamakan jumlah *successor* yang terbanyak, dimana elemen kerja tersebut diprioritaskan terlebih dahulu untuk ditempatkan dalam stasiun kerja dan diikuti oleh elemen kerja lain yang memiliki jumlah elemen kerja yang lebih sedikit. Apabila terdapat dua elemen yang memiliki bobot nilai yang sama, maka diprioritaskan pada elemen kerja yang memiliki waktu pengerjaan lebih besar. Apabila terdapat waktu pengerjaan yang sama. Pilihlah secara random elemen kerja yang diutamakan.
- 2.4. Lakukan penempatan elemen kerja yang memiliki bobot tertinggi hingga terendah ke setiap stasiun kerja.
- 2.5. Jika pada stasiun kerja terdapat kelebihan waktu, dalam hal ini waktu stasiun melebihi waktu siklus, pindahkan elemen pekerjaan yang ada dalam stasiun kerja tersebut ke stasiun kerja berikutnya selama tidak menyalahi *precedencediagram*.
- 2.6. Ulangi langkah 2.4 dan 2.5 sampai seluruh elemen kerja sudah ditempatkan ke dalam stasiun kerja.
- 2.7. Hitunglah *EL* dan *SI*.

Tahap 2. Local Search

Langkah 3

Inputkan data j, t_j, CT, EL, SI, ST dari inisial solusi yang telah terbentuk dan % *Th*.

Langkah 4

Set *current solusi* sebagai *intial solusi*.

Langkah 5

Set $k = 1$ dan $k_{maks} = n/2$.

Langkah 6

Set $e = 1$ dan $e_{maks} = m$.

Langkah 7

Tentukan *Idle Time*(*IT*) di setiap stasiun kerja dengan persamaan $IT = \{CT - ST_{ij}\}$.

Langkah 8

Pilih nilai IT_i terbesar (IT_{maks}) di setiap stasiun kerja dengan persamaan $IT_{maks} = \max_{i \in \{1, \dots, n\}} \{IT_i\}$ dan set sebagai NX_i . Jika terdapat lebih dari 1 stasiun kerja yang memiliki nilai IT_{maks} sama, maka pilihlah stasiun kerja secara random. Lanjutkan ke langkah 9.

Langkah 9

Periksa apakah $|j \in NX_i| = 1$?

Jika ya, set j yang terpilih sebagai VX dan lanjutkan ke langkah 11.

Jika tidak, lanjutkan ke langkah 10.

Langkah 10

Pilih $VX_j, j \in NX_i$ menggunakan persamaan $VX_j = \text{arg}_{j \in NX_i}^{\text{maks}} \{t_j\}$ kemudian set j dengan nilai terbesar sebagai VX_j . Jika terdapat nilai t_j yang sama, maka pilihlah secara random dan lanjutkan ke langkah 11.

Langkah 11

Lakukan proses *exchange* untuk semua VY_j . Pilih $VY_j, j \notin NX_i$ yang akan di *exchange* dan lanjutkan ke langkah 11.1

11.1. Apakah proses *exchange* melanggar *precedence constrain* atau VY_j sudah pernah menghasilkan konfigurasi yang sama?

Jika ya, lanjutkan ke langkah 11.2.

Jika tidak, lanjutkan ke langkah 11.3.

11.2. Apakah masih terdapat $j \notin NX_i$ yang belum ditukar?

Jika ya, kembali ke langkah 10.

Jika tidak, set $e = e + 1$, dan lanjut ke langkah 12.

11.3. Periksa apakah $ST_i > CT_i$?

Jika ya, kembali ke langkah 11.2.

Jika tidak, lanjut ke langkah 11.4.

11.4. Hitunglah $EL \& SI$ di setiap lintasan. Periksa apakah $EL \geq EL_{Th}$?

Jika ya, lanjut ke langkah 11.5.

Jika tidak, kembali ke langkah 11.2.

11.5. Apakah $|VY_j|$ dalam batas $EL \geq EL_{Th}$ $| > 1$?

Jika ya, pilih j dengan nilai EL terbesar, jika terdapat EL yang sama pilih nilai SI terkecil. Jika terdapat nilai SI yang sama pilihlah secara random dan lanjutkan ke langkah 11.6.

Jika tidak, lanjut ke langkah 11.6.

11.6. Apakah $EL = 100\%$?

Jika ya, lanjut ke langkah 16.

Jika tidak, ke langkah 13.

Langkah 12

Apakah $e > e_{\max}$?

Jika ya, lanjut ke langkah 13.

Jika tidak, kembali dan ulangi langkah 7.

Langkah 13

Set $i = 1$ sebagai N_i dan $i_{\max} = m$.

Langkah 14

Lakukan proses *insert* untuk j yang memungkinkan. Pilih $j \notin NI_i$ yang akan di *insert* ke NI_i dan lanjutkan ke langkah 14.1.

14.1. Apakah terdapat j yang dapat dilakukan proses *insert* melanggar *precedence constrain* terhadap NI_i atau apakah j yang di *insert* terhadap NI_i sudah pernah menghasilkan konfigurasi yang sama?

Jika tidak, lanjutkan ke langkah 14.2.

Jika ya, set $i = i + 1$, dan kembali ke langkah 14.3.

14.2. Periksa apakah $ST_i > CT_i$?

Jika ya, set $i = i + 1$ dan kembali ke langkah 14.3.

Jika tidak, lanjut ke langkah 14.4.

14.3. Apakah $i = m + 1$?

Jika ya, lanjut ke langkah 16.

Jika tidak, kembali dan ulangi langkah 14.

14.4. Apakah jumlah stasiun kerja berkurang?

Jika ya, set sebagai *current solution* dan kembali ke langkah 3.

Jika tidak, lanjut ke langkah 14.4.

14.5. Hitung nilai EL dan SI . Apakah $EL \geq EL_{Th}$?

Jika ya, lanjutkan ke langkah 14.6.

Jika tidak, $i = i + 1$, kembali ke langkah 14.3.

14.6. Apakah terdapat j hasil *insert* dalam batas $EL \geq EL_{Th}$?

Jika ya, pilih j dengan nilai EL terbaik. Jika terdapat nilai EL yang sama maka pilihlah nilai SI yang terbaik. Jika sama pilihlah secara random. Lanjutkan ke langkah 14.7.

Jika tidak, ke langkah 14.7.

14.7. Apakah $EL = 100\%$?

Jika ya, ke langkah 16.

Jika tidak, ke langkah 15.

Langkah 15

Pilihlah nilai EL terbaik dari Intial solusi, *insert*, *Insert* dan set konfigurasi EL terbaik sebagai *current solusi* untuk iterasi selanjutnya dan set $k = k + 1$. Kembali ke langkah 3.

Langkah 16

Apakah $k > n/2$?

Jika ya, lanjut ke langkah 17.

Jika tidak, kembali ke langkah 3.

Langkah 17

Berhenti, simpan dan tampilkan konfigurasi terbaik.

4. PENGUJIAN ALGORITMA DAN ANALISIS

Pengujian algoritma ini dilakukan untuk mengetahui performansi dari algoritma usulan yang telah dibuat dengan menggunakan dua skenario. Skenario 1 menggunakan data dari Su dan Lu (2007), Skenario 2 menggunakan data dari Bedworth dan Bailey (1982) dan Skenario 3 menggunakan Thomopoulos (1970).

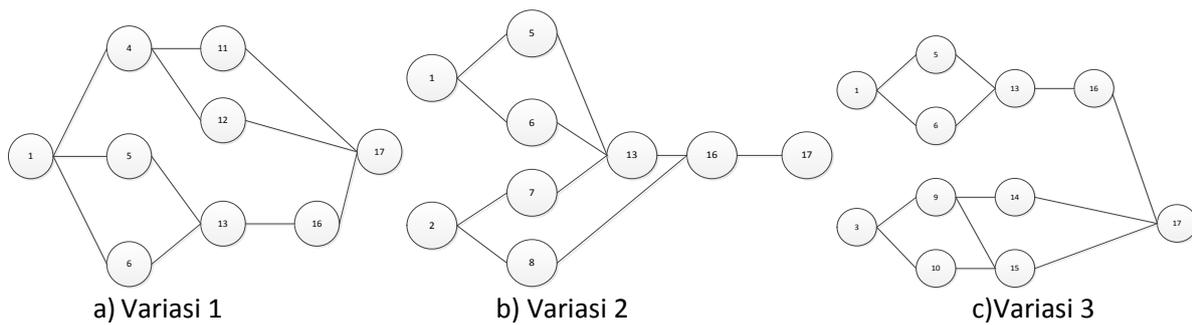
1. Set data Su dan Lu (2007)

Tabel waktu proses dan precedence diagram gabungan Su&Lu (2007) dapat dilihat pada Tabel 1 dan Gambar 3.

Tabel1. Tabel Waktu Proses Su&Lu (2007)

j	variasi (z)			Average
	z_1	z_2	z_3	
1	2	2	3	2.33
2	0	3	0	1.00
3	0	0	2	0.67
4	2	0	0	0.67
5	1	2	1	1.33
6	1	2	1	1.33
7	0	2	0	0.67
8	0	5	0	1.67
9	0	0	1	0.33
10	0	0	3	1.00
11	5	0	0	1.67
12	2	0	0	0.67
13	3	1	1	1.67
14	0	0	2	0.67
15	0	0	4	1.33
16	3	1	1	1.67
17	2	1	1	1.33

Algoritma Variable Neighborhood Descent with Fixed Threshold untuk Keseimbangan Lintasan Perakitan Mixed Model Dengan Kriteria Minimisasi Jumlah Stasiun Kerja



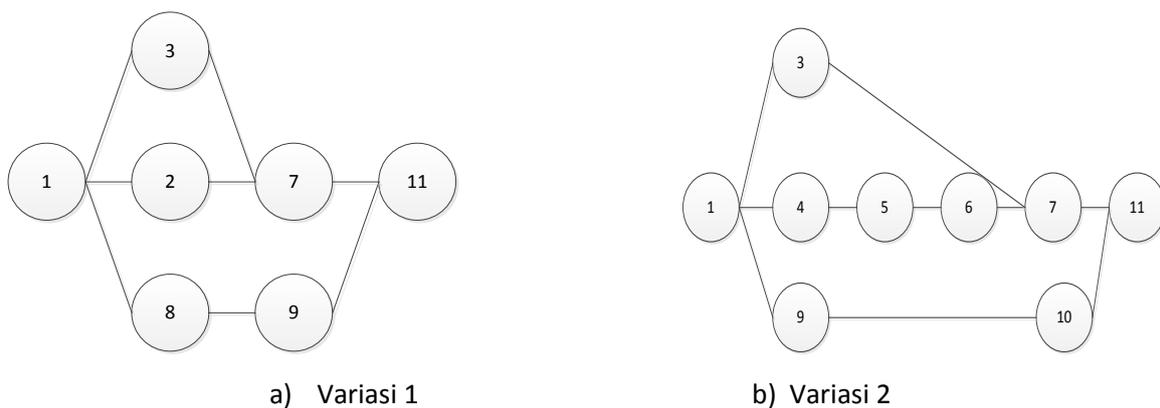
Gambar 3. Precedence Diagram Su & Lu (2007) untuk Variasi 1, 2, 3

2. Set data Bedworth dan Bailey (1982)

Set data dari Bedworth dan Bailey (1982) dapat dilihat pada Tabel 2 dan Gambar 4.

Tabel 2. Set Data Bedworth dan Bailey (1982)

<i>j</i>	variasi (z)		Average
	z_1	z_2	
1	1	1	1
2	5	0	2.5
3	4	4	4
4	0	1	0.5
5	0	5	2.5
6	0	6	3
7	2	2	2
8	4	0	2
9	3	3	3
10	0	5	2.5
11	3	3	3



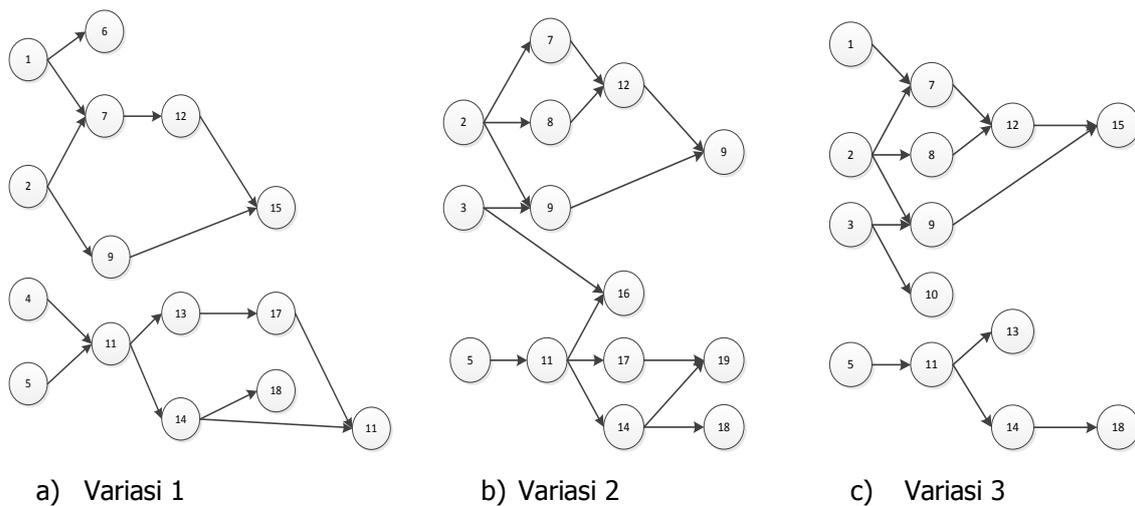
Gambar 4. Precedence Diagram Bedworth dan Bailey (1982) untuk Variasi 1, 2

3. Set data Thomopoulos (1970)

Set data penelitian Thomopoulos (1970) dapat dilihat pada Tabel 3 dan Gambar 5.

Tabel 3. Data Waktu Proses (t_j) Thomopoulos (1970)

j	variasi (z)			Average
	z_1	z_2	z_3	
1	0.5	0	1	0.50
2	0.4	0.8	1.2	0.80
3	0	0.2	0.4	0.20
4	0.4	0	0	0.13
5	0.2	0.2	0.2	0.20
6	0.2	0	0	0.07
7	0.4	0.5	0.6	0.50
8	0	0.5	0.5	0.33
9	0.4	0.3	0.2	0.30
10	0	0	0.2	0.07
11	0.3	0.3	0.3	0.30
12	0.1	0.3	0.5	0.30
13	0.1	0	0.1	0.07
14	0.2	0.2	0.2	0.20
15	0.7	1	1.5	1.07
16	0	0.1	0	0.03
17	0.5	0.5	0	0.33
18	0.3	0.5	0.3	0.37
19	0.4	0.3	0	0.23



b) Gambar 5. Precedence Diagram Thomopoulos (1970) untuk Variasi 1, 2, 3

Set data penelitian Su dan Lu (2007) yang terdiri dari 3 variasi dengan jumlah elemen kerja (n) = 17. Jumlah *demand* dari setiap variasi adalah $d1 = 40$, $d2 = 10$, $d3 = 30$ setiap *shift*.

Variasi 1

$$CT_i = \frac{\text{waktu kerja}}{\text{demand}}$$

$$CT_i = \frac{480}{40} = 12$$

Variasi 3

$$CT_i = \frac{\text{waktu kerja}}{\text{demand}}$$

$$CT_i = \frac{480}{30} = 16$$

Variasi 2

$$CT_i = \frac{\text{waktu kerja}}{\text{demand}}$$

$$CT_i = \frac{480}{10} = 48$$

Gabungan

$$CT = \frac{\text{waktu kerja}}{\text{demand}}$$

$$CT = \frac{480}{40+10+30} = 6$$

Tahap 1. Initial Solution

Langkah 1

Proses penggabungan *precedence diagram* dengan menggunakan *precedence matrices*. *Precedence matrices* setiap variasi dapat dilihat pada Gambar 6.

	1	4	5	6	11	12	13	16	17
1	0	1	1	1	1	1	1	1	1
4			0	0	1	1	0	0	1
5				0	0	0	1	1	1
6					0	0	1	1	1
11						0	0	0	1
12							0	0	1
13								1	1
16									1
17									

	1	2	5	6	7	8	13	16	17
1		0	1	1	0	0	1	1	1
2			0	0	1	1	1	1	1
5				0	0	0	1	1	1
6					0	0	1	1	1
7						0	1	1	1
8							1	1	1
13								1	1
16									1
17									

	1	3	5	6	9	10	13	14	15	16	17
1		0	1	1	0	0	1	0	0	1	1
3			0	0	1	1	0	1	1	0	1
5				0	0	0	1	0	0	1	1
6					0	0	1	0	0	1	1
9						0	0	1	1	0	1
10							0	0	1	0	1
13								0	0	1	1
14									0	0	1
15										0	1
16											1
17											

a) Variasi 1 b) Variasi 2 c) Variasi 3

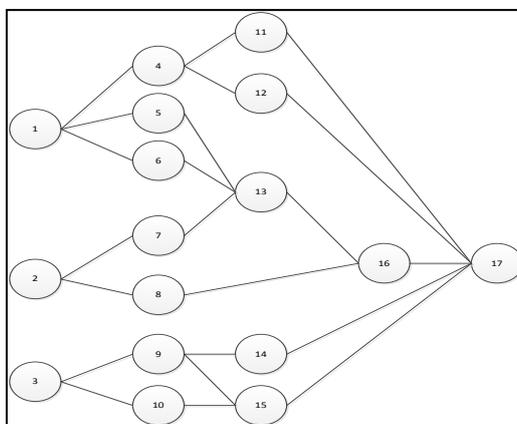
Gambar 6. Precedence Matrics Setiap Variasi

Setelah itu lakukan penggabungan *precedence matrices* untuk kedua variasi, sehingga didapat sebuah *precedence matrices* gabungan yang dapat dilihat pada Gambar 7.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1		0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1
2			0	0	0	0	1	1	0	0	0	0	1	0	0	1	1
3				0	0	0	0	0	1	1	0	0	0	1	1	0	1
4					0	0	0	0	0	0	1	1	0	0	0	0	1
5						0	0	0	0	0	0	0	1	0	0	1	1
6							0	0	0	0	0	0	1	0	0	1	1
7								0	0	0	0	0	1	0	0	1	1
8									0	0	0	0	0	0	0	1	1
9										0	0	0	0	1	1	0	1
10											0	0	0	0	1	0	1
11												0	0	0	0	0	1
12													0	0	0	0	1
13														0	0	1	1
14															0	0	1
15																0	1
16																	1
17																	

Gambar 7. Precedence Matrics Gabungan

Setelah melakukan *precedence matrices*, maka didapatkan *precedence diagram* gabungan dapat dilihat pada Gambar 8.



Gambar 8. precedence diagram gabungan

Perbandingan hasil perhitungan dengan model usulan untuk ketiga set data dapat dilihat pada Tabel 4, sedangkan hasil perbandingan model usulan dengan penelitian yang terpublikasi dapat dilihat pada Tabel 5.

Tabel 4. Perbandingan Hasil Akhir Agloritma Usulan

Set Data	% <i>Th</i>	Jumlah Elemen Kerja (<i>n</i>)	Waktu Siklus (<i>CT</i>)	Jumlah Stasiun kerja	EL (%)	SI
Bedworth & Bailey (1982)	5%	11	8.5	3	91,23%	2.06
	10%	11	9.5	3	91,23%	2.06
	15%	11	9.5	3	91,23%	2.06
Su & Lu (2007)	5%	17	5	4	100%	0.00
	10%	17	5	4	100%	0.00
	15%	17	5	4	100%	0.00
Thomopoulos (1990)	5%	19	2.13	3	93.75%	0.32
	10%	19	2.13	3	93.75%	0.32
	15%	19	2.13	3	93.75%	0.32

Tabel 5. Tabel Hasil Perbandingan Algoritma Usulan dengan Penelitian dipublikasi

Set Data	Optimasi	Heuristik	Metaheuristik	Jumlah Elemen Kerja (<i>n</i>)	Waktu Siklus (<i>CT</i>)	Jumlah Stasiun kerja	EL (%)	SI	
Bedworth & Bailey (1982)	Gokcen dan Erel (1998)			11	10	3	86.67%	2.06	
			Haq <i>et al.</i> (2006)		10				
			Yohana (2011)		10				
			Model Usulan		9.5				91,23%
Su & Lu (2007)			Su & Lu(2007)	17	6	4	83.33%		
			Yohana (2011)				83.33%	0.82	
			Model Usulan				5	100%	0.00
Thomopoulos (1970)		Thomopoulos (1970)		19	2.2	3	90.91%	0.07	
			Yohana (2011)	19			3	90.91%	0.07
			Model Usulan	19			2.13	3	93.75%

Berdasarkan Tabel 5 dapat dilihat hasil rekapitulasi akhir pada 3 set data menggunakan % nilai *threshold* yang berbeda. Setiap set data di ujiakan menggunakan nilai *threshold* sebesar 5%, 10%, 15%. Pada set data Bedworth & Bailey didapat hasil akhir yang berbeda antara nilai *threshold* sebesar 5% dengan 10% dan 15 %. Pada set data Bedworth & Bailey dengan nilai *threshold* sebesar 5 % menghasilkan stasiun kerja (SK) sebanyak 4 SK dengan *CT* sebesar 8.5 sedangkan untuk set data Bedworth & Bailey dengan nilai *threshold* sebesar 10 % dan 15% menghasilkan stasiun kerja (SK) sebanyak 3 SK dengan *CT* sebesar 9.5 hal ini dikarenakan perbedaan batas ambang *threshold* akan mempengaruhi *range* nilai yang dapat diterima. Sehingga akan mempengaruhi hasil akhir yang didapat.

Pada set data Su & Lu didapat hasil akhir yang sama antara nilai *threshold* sebesar 5% dengan 10% dan 15 % yaitu menghasilkan stasiun kerja (SK) sebanyak 4 SK dengan *CT* sebesar 5. Pada set data Thomopoulos didapat hasil akhir yang sama antara nilai *threshold* sebesar 5% dengan 10% dan 15 % yaitu menghasilkan stasiun kerja (SK) sebanyak 3 SK dengan *CT* sebesar 2.13.

5. KESIMPULAN & SARAN

1. Algoritma keseimbangan lintasan perakitan yang dihasilkan adalah algoritma yang digunakan untuk menyelesaikan permasalahan keseimbangan lintasan perakitan *mixed model* menggunakan algoritma *variable neighborhood descent with fixed threshold* dengan kriteria minimisasi jumlah stasiun kerja.
2. Pada pengujian algoritma menggunakan set data Bedworth & Bailey didapat hasil yang berbeda disetiap nilai *Threshold*. Hal ini menunjukkan setiap nilai *Threshold* memiliki ambang batas penerimaan yang berbeda. Semakin kecil nilai *Threshold* maka batas penerimaan semakin kecil (ketat).

Saran yang ditunjukkan untuk penelitian selanjutnya adalah melakukan rancangan keseimbangan lintasan perakitan *mixed model* dengan menggunakan algoritma *Threshold Accepting* dimana nilai *threshold* yang digunakan tidak ditentukan secara tetap dan dibuat dengan menggunakan program.

REFERENSI

- Dueck, G. and Scheuer, T. (1990). Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing. *Jurnal of Computation Physics* 90. 161-175
- Elsayed, E. A., dan Boucher, T. O., 1994, *Analysis and Control of Production Systems*. Second Edition. Prentice Hall New Jersey.
- Gökçen, H., Erel, E., 1998, Binary Integer Formulation For Mixed-Model Assembly Line Balancing Problem. *Computers & Industrial Engineering* 34, 451-461.
- Hansen P. dan Mladenovic., 1999, An Introduction to Variable Neighbourhood Search. In S.Voss, S. Martello, L. H. Osman, and C. Roucairol, editors, *Meta Heuristics*, Kluwer Academic Publisher.
- Imran, A., dan Okdinawati, L., 2010, A Threshold Accepting Heuristik For The Heterogeneous Fleet Vehicle Routing Problem. *Jurnal of business logictic*, 15-26.
- Scholl, A., 1999, *Balancing and Sequencing of Assembly Lines*. Second Edition. Physica-Verlag Heidelberg New York.
- Thomopoulos, N. T., 1970, Mixed Model Line Balancing With Smoothed Station Assignments. *Management Science*, 16, 593-603.