

Model Keseimbangan Lintasan Perakitan *Mixed Model* Menggunakan Algoritma *Guided Greedy Randomized Adaptive Search Procedures* dengan Kriteria Minimisasi Jumlah Stasiun Kerja

Alex Saleh, Emsosfi Zaini, Puji Purwaning Rahayu
Laboratorium Sistem Produksi, Jurusan Teknik Industri,
Institut Teknologi Nasional (Itenas) – Bandung
Email : alex@itenas.ac.id

ABSTRAK

Penelitian ini membahas model keseimbangan lintasan perakitan mixed model menggunakan algoritma guided greedy randomized adaptive search procedures (Guided GRASP) dengan kriteria minimisasi jumlah stasiun kerja. Keseimbangan lintasan perakitan mixed model adalah model lintasan perakitan tunggal yang merakit lebih dari satu produk sejenis atau memiliki karakteristik yang sama. Guided GRASP merupakan pendekatan metaheuristik yang terdiri dari dua tahap yaitu tahap inisial solusi dan local search. Tahap 1 adalah tahap pembentukan solusi inisial yang dibentuk melalui restricted candidate list (RCL). RCL disusun menggunakan suatu fungsi greedy berbasis penambahan waktu operasi dari setiap elemen kerja yang akan ditempatkan dan suatu threshold parameter α . Tahap 2 bertujuan untuk mengurangi jumlah stasiun kerja sehingga efisiensi lintasan yang dihasilkan besar. Pada Tahap 2, dilakukan peneksplorasi solusi dengan cara menukar elemen kerja (exchange) dan menambahkan elemen kerja di stasiun kerja lain (insert) dengan tetap tidak melebihi waktu siklus dan tidak melanggar precedence relations. Performansi model usulan diuji dengan menggunakan data-data dari literatur. Hasil pengujian menunjukkan bahwa model usulan dapat memberikan solusi yang sama dengan penelitian-penelitian sebelumnya yang telah dipublikasikan.

Kata kunci: lintasan perakitan, heuristik, guided GRASP, mixed model, jumlah stasiun kerja

ABSTRACT

This study discusses a model assembly line balancing mixed models using randomized greedy algorithm guided adaptive search procedures (Guided GRASP) with minimization criteria the number of work stations. Mixed assembly line balancing model is a model of a single assembly line to assemble more than one product type or have the same characteristics. Guided GRASP is a metaheuristic approach which consists of two phases: the initial solution and local search. Phase 1 is the initial stage of the formation of a solution formed by the restricted candidate list (RCL). RCL were prepared using a function -based greedy additional operation time of each element of the work that will be placed and a threshold parameter α . Phase 2 aims to reduce the number of work stations so that the efficiency of the resulting trajectory great. In Phase 2, conducted exploring a solution by swapping elements of work (exchange) and add an element of work in another work station (insert) while not exceeding the cycle time and does not violate the precedence relations. The performance of the proposed model was tested using data from the literature. The results show that the proposed model can provide a solution similar to previous studies that have been published.

Keywords: assembly line, heuristic, guided GRASP, mixed model, the number of work stations

1. PENDAHULUAN

Lintasan perakitan dapat didefinisikan sebagai satu set stasiun kerja ($i=1,2,3,\dots,m$) yang dihubungkan menggunakan suatu sistem *material handling*. Satu set elemen kerja ($j=1,2,3,\dots,n$) ditempatkan pada satu set stasiun kerja untuk dirakit dengan memperhatikan hubungan preseden yang diwakili oleh *precedence diagram*. Penugasan satu set elemen kerja terhadap satu set stasiun kerja dengan tidak melanggar hubungan preseden dan pembatas waktu siklus dikenal sebagai permasalahan.

Permasalahan keseimbangan lintasan perakitan sering juga disebut dengan *simple assembly line balancing problem* (SALBP). Banyak penelitian yang membahas SALBP diantaranya Talbot *et al.* yang meneliti empat prosedur heuristik berbasis aturan prioritas *single pass* dan prosedur *backtracking* untuk meminimisasi jumlah stasiun kerja [1]. Hackman *et al.* mengusulkan algoritma *branch and bound* (B&B) yang dikombinasikan dengan beberapa aturan *fathoming* heuristik untuk mengurangi jumlah percabangan [2]. Ponnambalam *et al.* melakukan studi komparatif terhadap enam pendekatan heuristik berbasis heuristik konstruktif dalam menyelesaikan SALBP untuk meminimisasi jumlah stasiun kerja [3].

Scholl dan Voß mengajukan pendekatan metaheuristik berbasis *Tabu Search* (TS) dalam pencarian solusi SALBP dengan kriteria minimisasi jumlah stasiun kerja [4]. Beberapa penelitian mengusulkan pendekatan modern heuristik (metaheuristik) untuk menyelesaikan SALBP diantaranya Goncalves dan De Almeida mengusulkan algoritma genetik campuran dengan kriteria minimisasi jumlah stasiun kerja. Bautista dan Pereira menerapkan *ant colony optimization* (ACO) untuk SALBP dengan kriteria minimisasi jumlah stasiun kerja [5], [6]. Lapierre *et al.* menerapkan algoritma TS untuk penyelesaian SALBP dengan kriteria minimisasi jumlah stasiun kerja [7]. Andrés *et al.* membandingkan penyelesaian SALBP dengan menggunakan model matematis *binary linear program* (BLP) dan GRASP dengan kriteria minimisasi jumlah stasiun kerja [8]. Penelitian di atas hanya membahas *assembly line* untuk produk tunggal. Bila dalam satu lintasan dirakit lebih dari satu produk, maka lintasan tersebut disebut *mixed model assembly line balancing* (MALB).

Masalah MALB termasuk komplikasi sebagai akibat dari *blocked* dan *starvation* yang disebabkan oleh kedatangan model yang berbeda ke lintasan yang memiliki waktu perakitan yang berbeda (Bukchin, [9]). Model yang diusulkan untuk dapat memecahkan permasalahan MALB diantaranya *Binary integer programming* dan *goal programming* (Gokcen & Erel [10]). Algoritma pada dasarnya dikembangkan untuk meminimalkan tingkat pekerjaan yang tidak lengkap sehingga dapat mengurangi *work in process*.

Pada penelitian ini digunakan metode *Guided Greedy Randomized Adaptive Search Procedures* (*Guided GRASP*). *Greedy Randomized Adaptive Search Procedures* (GRASP) adalah metode metaheuristik untuk memecahkan masalah optimisasi kombinatorial. Pertama kali GRASP diusulkan oleh Feo dan Resende untuk menyelesaikan masalah *travelling salesman problem* dan penjadwalan pengemudi bus [11]. GRASP merupakan kombinasi dari *greedy heuristic*, pemilihan acak, dan prosedur pencarian lokal. GRASP terdiri dari dua tahap yaitu tahap konstruksi untuk mendapatkan solusi inisial dan tahap *local search* yang berfungsi untuk memperbaiki solusi tahap konstruksi. GRASP telah banyak diterapkan untuk menyelesaikan masalah-masalah optimisasi kombinatorial seperti penentuan lokasi, *routing*, penjadwalan, jaringan telekomunikasi, penugasan, dan sebagainya. Studi literatur mengenai GRASP dan aplikasinya dapat dilihat pada Resende dan Ribiero [12], [13]. Selanjutnya dikembangkan Algoritma *Guided GRASP* oleh Luis [14].

Memperhatikan hal-hal di atas, maka perlu dikembangkan algoritma *Guided GRASP* untuk menyelesaikan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja.

2. PENGEMBANGAN MODEL

Pengembangan model pada penelitian ini dijelaskan dalam dua tahap, yaitu model dasar dan model usulan.

2.1 Model Dasar

Model dasar untuk penelitian ini merupakan pengembangan dari beberapa penelitian diantaranya adalah Andrés *et al.* yang menyelesaikan *simple assembly line balancing problem* (SALBP) menggunakan GRASP dengan kriteria minimisasi jumlah stasiun kerja [8]. Tujuan pengembangan model ini adalah untuk mendapatkan rancangan lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja menggunakan model usulan yang diadaptasi dari *greedy randomized adaptive search procedure* (GRASP).

2.2 Model Usulan

Pengembangan metode *greedy randomized adaptive search procedures* (GRASP) saat ini sudah sangat banyak diantaranya adalah *Guided GRASP*, *Reactive GRASP*, *Parallel GRASP*, *Multi Level GRASP*, dan *Porturbation GRASP*. Pengembangan model usulan yang digunakan pada penelitian kali ini adalah *Guided GRASP*. *Guided GRASP* merupakan gabungan dari *Reactive GRASP* dan *Retricted Region*. Pada *Reactive GRASP* parameter α yang digunakan tidak diambil secara random melainkan bisa diganti dengan fungsi linier atau fungsi *convex and concave*. Sedangkan *Retricted Region* parameter α ditentukan secara random dari *area rectangular* pada suatu wilayah perencanaan. *Guided GRASP* akan menghasilkan solusi lebih baik jika ditambahkan fungsi lain misalnya pada model usulan ini fungsi yang digunakan adalah delta.

Guided greedy randomized adaptive search procedures terbagi menjadi dua tahapan yaitu konstruksi solusi inisial dan pencarian lokal. Notasi-notasi yang digunakan pada model usulan adalah sebagai berikut:

i	indeks untuk stasiun kerja (SK); ($i = 1, 2, \dots, m$)
z	indeks untuk variasi produk; ($z = 1, 2, \dots, Z$)
t_j	waktu proses pada elemen kerja j
d_j	<i>demand</i> untuk setiap variasi produk
CT	waktu siklus
k	indeks iterasi pada <i>local search</i> ; ($k = 1, 2, \dots, k_{max}$)
p	indeks iterasi keseluruhan ($p = 1, 2, \dots, p_{max}$)
α	<i>threshold parameter</i> ; $\alpha \in U[0.5, 1]$
V	himpunan elemen kerja yang dapat ditambahkan (<i>assignable</i>)
V_c	himpunan elemen kerja yang dapat dijadikan kandidat prospektif pemilihan RCL
V_a	himpunan elemen kerja yang telah ditambahkan ke dalam SK (<i>assigned</i>)
ST_i	akumulasi waktu elemen kerja pada SK i
ST_{iz}	akumulasi waktu elemen kerja pada SK i pada setiap variasi produk
Δ	selisih waktu siklus dengan akumulasi waktu elemen kerja pada SK i
EL_k	efisiensi lintasan pada iterasi k
SI_k	<i>smoothness index</i> pada iterasi k
EL_p	efisiensi lintasan pada iterasi p
SI_p	<i>smoothness index</i> pada iterasi p
$h(j)$	nilai fungsi <i>greedy</i> setelah penambahan elemen $j \in RCL$
$\underline{h}(j)$	total waktu SK minimum setelah penambahan elemen kerja ke- j
$\overline{h}(j)$	total waktu SK maksimum setelah penambahan elemen kerja ke- j

Algoritma usulan untuk menyelesaikan masalah *mixed model assembly line balancing* memiliki tahapan-tahapan sebagai berikut:

1. Tahap 1 – Initial Solution

Tahap ini merupakan tahap pembentukan solusi inisial. Solusi inisial berasal dari pembentukan *restricted candidate list* (RCL) yang disusun berdasarkan suatu fungsi *greedy*. Nilai *threshold parameter* α ditentukan nilainya sebagai suatu besaran yang telah ditetapkan. Pada penelitian ini, nilai α dibangkitkan secara *random* dari $U[0.5,1]$ dengan *step size* 0.1. Satu per satu elemen pada RCL dipilih secara *guided* dengan pemilihan delta *idle time* terkecil sampai solusi inisial terbentuk secara lengkap. Solusi inisial yang terbentuk merupakan rancangan lintas perakitan lurus (*simple line balancing*).

Pada proses pembentukan RCL, tidak semua elemen pekerjaan dapat dijadikan kandidat prospektif V_c . Pemilihan elemen kerja dilakukan berdasarkan *precedence relations* yang dilihat pada *precedence diagram* serta batasan waktu siklus (*CT*). Dalam penentuan selang RCL digunakan nilai maksimum dan minimum dari waktu stasiun itu sebagai berikut:

a. Nilai total stasiun kerja minimum

$$\underline{h}(j) = \min(\sum_{j \in V_a} t_j + t_{j \in V_c}) \quad (1)$$

b. Nilai total stasiun kerja maksimum

$$\bar{h}(j) = \max(\sum_{j \in V_a} t_j + t_{j \in V_c}) \quad (2)$$

Sedangkan RCL disusun dengan menggunakan persamaan:

$$RCL = \{ \underline{h}(j) \leq h(j) \leq \bar{h}(j) + \alpha(\bar{h}(j) - \underline{h}(j)) \} \quad (3)$$

Pada persamaan (3) $h(j)$ merupakan waktu elemen kerja yang berada diantara $\underline{h}(j)$ dan $\bar{h}(j)$ Untuk menjamin jumlah elemen pada RCL tidak berjumlah nol, maka diusulkan bahwa $|RCL| \geq L_{min}$ dengan nilai L_{min} adalah:

$$L_{min} = \max\left(2, \min\left(\left\lceil \frac{n}{6} \right\rceil, |V_c|\right)\right) \quad (4)$$

Jika nilai $|RCL| \geq L_{min}$ maka nilai parameter α dilonggarkan dengan mengizinkan sejumlah elemen yang telah diurutkan dalam V_c dari yang terkecil ke terbesar untuk dimasukkan ke dalam RCL. Proses ini disebut proses *relaxing*.

2. Tahap 2 – Local Search

Tahap ini merupakan tahap eksplorasi terhadap pembentukan solusi inisial yang telah dibentuk pada Tahap 1 sehingga dapat dihasilkan solusi optimal yang dipilih sebagai solusi akhir. Tahap 2 ini bertujuan untuk mengurangi jumlah stasiun kerja sehingga efisiensi lintasan yang dihasilkan besar. Seluruh tahap ini dilakukan beberapa kali (iteratif) sehingga *local optimum* terbaik dipilih sebagai solusi akhir. Jumlah iterasi maksimum untuk mengeksplorasi solusi dinyatakan dengan $k_{max} = \max\left(4, \left\lceil \frac{n}{4} \right\rceil\right)$. Pada penelitian ini pengeksplorasi solusi dilakukan dengan cara menukar elemen kerja (*exchange*) dan menambahkan elemen kerja di stasiun kerja lain (*insert*) dengan tetap tidak melebihi waktu siklus dan tidak melanggar *precedence relations*.

Untuk mengukur hasil rancangan lintasan perakitan maka dilakukan pengukuran utilisasi diantaranya adalah efisiensi lintasan (EL_k) yang menyatakan tingkat efisiensi stasiun kerja rata-rata pada suatu lintasan perakitan. Pada iterasi ke- k EL_k dihitung menggunakan persamaan:

$$EL_k = \frac{\sum_{i=1}^m ST_i}{m \times CT} \times 100\% \quad (5)$$

Bila dalam iterasi k ditemukan $EL_k = EL_{k-1}$ maka ada perhitungan utilisasi lainnya itu *smoothness index* (SI_k) yang mengindikasikan kelancaran relatif dari sebuah lintas perakitan. SI_k dihitung menggunakan persamaan:

$$SI_k = \sqrt{\sum_{i=1}^m (ST_{max} - ST_i)^2} \quad (6)$$

Dalam penelitian ini, pengulangan iterasi secara keseluruhan dari Tahap 1 sampai Tahap 2 dilakukan dengan batasan $p_{max} = \max\left(2, \left\lceil \frac{n}{10} \right\rceil\right)$. Bila nilai p_{max} telah tercapai, maka pilih solusi optimal akhir dengan kriteria maksimisasi EL_k . Bila perbandingan EL_k masih sama, pilih minimisasi SI_k .

2.3 Algoritma Model Usulan

Algoritma usulan *Guided* GRASP untuk menyelesaikan masalah lintasan perakitan *mixed model* dengan kriteria meminimisasi jumlah stasiun kerja adalah sebagai berikut:

Tahap 1 – *Initial Solution*

Langkah 0

Input data: $i, j, t_j, d, CT, V, V_c = \emptyset$, *precedence diagram* masing-masing variasi.

Langkah 1

Lakukan proses penggabungan *precedence diagram* dengan menggunakan *precedence matrices*.

Langkah 2

set $p_{max} = \max\left(2, \left\lceil \frac{n}{10} \right\rceil\right)$

Langkah 3

Set $p = 1$ dan Set $i = 1$

Langkah 4

a. Perbaharui kandidat prospektif (V_c)

Pilih elemen kerja $j \in V$ yang dapat dijadikan kandidat untuk ditambahkan ke dalam SK ke- i dengan memperhatikan *precedence relations*.

b. Periksa apakah $V_c = 1$?

Jika ya, lanjutkan ke Langkah 6; dan tempatkan elemen kerja j ke dalam V_a pada SK ke- i .

Lainnya, lanjutkan ke Langkah 5.

Langkah 5

a. Tentukan ST_i minimum dengan Persamaan (1) dan maksimum dengan Persamaan (2)

b. Bangkitkan secara random $\alpha \in U[(0,5), 1]$

c. Bentuk RCL dengan menggunakan Persamaan (3)

d. Set L_{min} dengan menggunakan Persamaan (4)

e. Periksa apakah $RCL < L_{min}$?

Jika ya, lakukan proses *relaxing*; urutkan $j \in V_c$ dari t_j terkecil ke terbesar, pilih urutan sebanyak L_{min} untuk masuk ke dalam RCL dan lanjutkan ke Langkah 6

Lainnya, lanjutkan ke Langkah 6.

Langkah 6

Hitung ST_{iz} setiap elemen kerja j pada *RCL* dan periksalah apakah $ST_{iz} > CT$ bila ditambahkan dengan elemen kerja tersebut?

Jika ya, perbaiki *RCL* dan lanjutkan ke Langkah 7;

Lainnya lanjutkan ke Langkah 7.

Langkah 7

Hitung $\Delta = CT - ST_i$ setiap elemen kerja j pada *RCL* kemudian pilih elemen kerja dengan nilai Δ terkecil dan lanjutkan ke Langkah 8.

Langkah 8

Periksa apakah $ST_i < CT$ dan atau $ST_m < CT$?

Jika ya, tempatkan elemen kerja j ke dalam V_a pada SK ke- i ; lanjutkan ke Langkah 9.

Lainnya, set $i = i + 1$; lalu tempatkan elemen kerja j ke dalam V_a pada SK ke- i lanjutkan ke Langkah 9.

Langkah 9

Periksa apakah $|V| = \emptyset$?

Jika ya, berhenti; lanjutkan ke Tahap 4 – *local search*.

Lainnya, kembali ke Langkah 4.

Tahap 2 – *Local Search*

Langkah 10

Input data: jalur lintasan perakitan pada Tahap 1, set $k_{max} = \max\left(4, \left\lceil \frac{n}{4} \right\rceil\right)$

Langkah 11

Set $k = 1$

Langkah 12

a. Lakukan proses *exchange*.

a.1. Ambil dua buah elemen kerja j dari SK ke- i yang berbeda secara sembarang.

a.2. Tukarkan kedua elemen kerja j tersebut.

Periksa apakah penukaran elemen kerja j melanggar preseden dan ST_i melebihi CT ?

Jika ya, batalkan, kembali ke Langkah 12a.1.;

Lainnya, tempatkan elemen kerja j pada SK ke- i , periksa apakah proses *exchange* masih bisa dilakukan?

Jika ya, batalkan, kembali ke Langkah 12a.1.;

Lainnya, lanjutkan ke Langkah 12b.

b. Lakukan proses *insert*.

b.1. Ambil salah satu elemen kerja j dari SK ke- i secara sembarang.

b.2. Pindahkan elemen kerja j ke SK_l ($l \neq i$)

Periksa apakah pemindahan elemen kerja j melanggar preseden dan ST_i melebihi CT ?

Jika ya, batalkan; kembali ke Langkah 12b.1.;

Lainnya, tempatkan elemen kerja j pada SK ke- i , periksa apakah proses *insert* masih dapat dilakukan?

Jika ya, kembali ke Langkah 12b.1.;

Lainnya, lanjutkan ke Langkah 13.

Langkah 13

Hitung efisiensi lintasan (EL_k) menggunakan Persamaan (5)

Langkah 14

Periksa apakah $EL_k = 100\%$?

Jika ya, lanjutkan ke Langkah 15

Lainnya, periksa apakah $k = k_{max}$?

simpan nilai EL_k lalu set $k = k + 1$ dan kembali ke Langkah 12.

Langkah 15

Pilih $\max EL_k = \max \{EL_k\}$ dan periksa apakah maksimum $|EL_k| > 1$?

Jika ya, pilih maksimum EL_k , lanjutkan ke Langkah 16

Lainnya, lanjutkan ke Langkah 16

Langkah 16

Hitung *Smoothness Index* (SI) menggunakan Persamaan (6) dan periksa apakah $|SI_k| > 1$?

Jika ya, pilih minimum SI_k lanjutkan ke Langkah 17.;

Lainnya, lanjutkan ke Langkah 17.

Langkah 17

Tampilkan rancangan lintasan SK perakitan *mixed model* terpilih beserta nilai EL_k , lalu set $\max EL_k = EL_p$ dan lanjutkan ke Langkah 18.

Langkah 18

a) Periksa apakah $p = p_{max}$?

Jika ya, tampilkan nilai EL_p dan pilih nilai $\max EL_p = \max \{EL_p\}$ kemudian periksa apakah maks $|EL_p| > 1$?

Jika ya, hitung SI_p menggunakan persamaan (6) dan lanjutkan ke Langkah 18b.;

Lainnya, lanjutkan ke Langkah 19.

Lainnya, Simpan nilai EL_p dan SI_p kemudian set $p = p + 1$ dan kembali ke Langkah 4.

b) Pilih min SI_p dan periksa apakah $|SI_p| > 1$?

Jika ya, pilih secara random min SI_p kemudian lanjutkan ke Langkah 19;

Lainnya, lanjutkan ke Langkah 19.

Langkah 19

Tampilkan lintasan sebagai *final local optimal* untuk MALB beserta parameternya.

3. PENGUJIAN MODEL DAN ANALISIS

Pengujian model bertujuan untuk menguji keandalan model usulan dengan hasil dari penelitian-penelitian sebelumnya. Data yang digunakan adalah data dari penelitian Bedworth dan Bailey dengan jumlah elemen kerja $n = 11$ [15], Thomopoulos dengan jumlah elemen kerja $n = 19$ [16], dan Su dan Lu dengan jumlah elemen kerja $n = 17$ [17].

Perbandingan jumlah stasiun kerja yang dihasilkan dari penelitian ini dan data-data yang digunakan dapat dilihat pada Tabel 1.

Tabel 1 Perbandingan Hasil Pengujian Model

Set Data	Jumlah Elemen Kerja (n)	Waktu Siklus (CT)	Jumlah Stasiun Kerja (SK)			
			Haq <i>et al</i> [18]	Thomopoulos [16]	Su dan Lu [17]	Model Usulan
Bedworth dan Bailey [15]	11	10	3			3
Thomopoulos [16]	19	2,2		3		3
Su dan Lu [17]	17	6			4	4

Berdasarkan Tabel 1 model *Guided GRASP* memiliki keandalan dalam merancang lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja. Pada penelitian Haq *et al* [18], set data Bedworth dan Bailey diselesaikan menggunakan metode *hybrid genetic approach (Hybrid GA)* [15]. Model usulan *Guided GRASP* dapat menghasilkan jumlah stasiun kerja yang sama sebanyak 3 stasiun kerja untuk set data Bedworth dan Bailey dengan jumlah elemen kerja (n) = 11 dan $CT = 10$ [15].

Model usulan *Guided GRASP* dapat menghasilkan jumlah stasiun kerja yang sama sebanyak 3 stasiun kerja untuk set data Thomopoulos dengan jumlah elemen kerja (n) = 19 dan $CT = 2,2$ [16]. Penelitian Thomopoulos menggunakan set data Thomopoulos diselesaikan dengan metode *mixed model line balancing with smoothed station assignments* [16].

Penelitian Su dan Lu diselesaikan dengan metode *genetic algorithm (GA)*, sedangkan dengan menggunakan model usulan *Guided GRASP* dapat menghasilkan jumlah stasiun kerja yang sama, yaitu sebesar = 4 dengan jumlah elemen kerja (n) = 17 dan $CT = 6$ [17].

4. SIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat diperoleh simpulan sebagai berikut:

1. Perancangan lintasan yang dihasilkan adalah model lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja. Model ini dikembangkan menggunakan algoritma *Guided Greedy Randomized Adaptive Search Procedures (Guided GRASP)*.
2. Pengujian model menunjukkan bahwa model usulan dapat menghasilkan hasil yang sama dengan set data Bedworth dan Bailey dengan jumlah stasiun kerja (n) = 3 [15], dan dengan set data Thomopoulos memperoleh hasil yang sama dengan jumlah stasiun kerja (n) = 3 [16], serta dengan set data Su dan Lu memperoleh hasil yang sama dengan jumlah stasiun kerja (n) = 4 [17].

3. Saran untuk pengembangan penelitian selanjutnya adalah melakukan perancangan lintasan perakitan *mixed model* dengan menggunakan metode lain yang belum pernah digunakan seperti algoritma *tabu search* dan *ant colony systems*. Pengembangan penelitian selanjutnya dapat juga dilakukan dengan membuka salah satu asumsi, sehingga sistem yang dibahas akan berbeda dengan penelitian ini.

DAFTAR PUSTAKA

- [1] Talbot, F.B., Gehrlein, W. V. dan Patterson, J. H., 1986, Comparative Evaluation of Heuristics Line Balancing Techniques. *Management Science*, 32, 430-454.
- [2] Hackman, S. T., M. J. Magazine dan T. S. Wee. 1989, Fast, Effective Algorithms for Simple Assembly Line Balancing Problems. *Operations Research*, 32, 916-924.
- [3] Ponnambalam, S. G., Aravindan, P., dan Naidu, G. M., 1999, A Multi-objective Genetic Algorithm for Solving Assembly Line Balancing Problem. *International Journal of Advance Manufacturing Technology*, 16, 341-352
- [4] Scholl, A., dan Voß, S., 1996, Simple Assembly Line Balancing – Heuristic Approaches. *Journal of Heuristics*, 2, 217-244.
- [5] Goncalves, J. F., dan De Almeida, J. R., 2002, A Hybrid Genetic Algorithm for Assembly Line Balancing. *Journal of Heuristics*, 8, 629-642.
- [6] Bautista, J., dan Pereira, J., 2002, Ant Algorithms for Assembly Line Balancing. *Lecture Notes in Computer Science*, 2463, 67-75. (Ant algorithms for a time and space constrained assembly line balancing problem)
- [7] Lapierre, S. D., Ruiz, A., dan Soriano, P., 2006, Balancing Assembly Lines with Tabu Search. *European Journal of Operational Research*, 168, 826-837.
- [8] Andrés, C., Miralles, C., dan Pastor, R., 2008, Balancing and Scheduling Tasks in Assembly Lines with Sequence-dependent Setup Times. *European Journal of Operational Research*, 187, 1212-1223.
- [9] Bukchin, J., Dar-El, E. M., dan Rubinovitz, J., 2001, Mixed Model Assembly Line Design in a Make-To-Order Environment. *Computers & Industrial Engineering*, 41, 405-421.
- [10] Gokcen, H., dan Erel, E., 1998, Binary Integer Formulation for Mixed-model Assembly Line Balancing Problem. *Computers ind. Engng*, 34, 451-461.
- [11] Feo, T. A., Resende, M. G. C., 1995, Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6, 109-133.
- [12] Resende, M. G. C., Ribiero, C. G., 2003, Greedy Randomized Adaptive Search Procedures. Dalam Glover, F. dan Kochenberger, G. A., editor. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 219-249.
- [13] Resende, M. G. C., Ribiero, C. G., 2005, Parallel Greedy Randomized Adaptive Search Procedures. Dalam Alba, E., editor. *Parallel Metaheuristics*. John Wiley and Sons, 315-346.
- [14] Luis, M., 2008, Meta-heuristics for the Capacitated Multi-Source Weber Problem. *Kent Business School University of Kent United Kingdom*, 26-28, 134-136.
- [15] Bedworth, D, D, and Bailey,., *Integrated Production Control Systems*. Wiley, New York, 1982, pp. 281-293.
- [16] Thomopoulos, N. T., 1970, Mixed Model Line Balancing with Smoothed Station Assignments, *Management Science*, 16, 593-603.
- [17] Su, P., dan Lu, Y., 2007, Combining Genetic Algorithm and Simulation for the Mixed Model Assembly Line Balancing Problem.
- [18] Haq, A. N., Jayaprakash, J, dan Rengarajan, K., 2005, A Hybrid Genetic Algorithm Approach to Mixed Model Assembly Line Balancing. *Int J Adv Manuf Thecnol*, 28, 337-341.