

Algoritma *Guided Greedy Randomized Adaptive Search Procedures with Fixed Threshold* untuk Keseimbangan Lintasan Perakitan U-Shaped dengan Kriteria Minimisasi Jumlah Stasiun Kerja*

KHAIRAN ADI PUTRA, ARIF IMRAN, EMSOSFI ZAINI.

Jurusan Teknik Industri
Institut Teknologi Nasional Bandung
Email : khairanadiputra@gmail.com

ABSTRAK

Assembly line balancing (ALB) adalah suatu penugasan penempatan pekerjaan pada stasiun kerja yang saling berhubungan antar lintasan produksi untuk memenuhi target laju produksi. Pada penelitian ini akan dibahas keseimbangan lintasan perakitan U-shaped dengan kriteria minimisasi jumlah stasiun kerja. Untuk menyelesaikan permasalahan tersebut dikembangkan algoritma Guided GRASP with Fixed Threshold. Pengujian algoritma usulan dilakukan dengan menggunakan tiga skenario, Skenario 1 menggunakan set data Scholl (1999) dengan $n = 12$, Skenario 2 menggunakan set data Jackson (1956) dalam Rahayuningtyas (2011) dengan $n = 11$, Skenario 3 menggunakan set data Baykasoglu (2006) dengan $n = 12$. Skenario 1 menghasilkan solusi yang sama dengan hasil yang telah dipublikasikan, sedangkan skenario 2 dan 3 menghasilkan solusi yang lebih baik dari penelitian yang telah dilakukan sebelumnya.

Kata Kunci: Keseimbangan Lintasan, U-shaped, Guided GRASP, Fixed Threshold

ABSTRACT

Assembly line balancing (ALB) is a job placement assignment on workstations that are interconnected across the line to meet the production rate. The discussion carried out in this study is about the balance of the line assembly using U-shaped with a minimization criterion number of work stations. The Guided GRASP with Fixed Threshold algorithm is developed to tackle this problem. The proposed algorithm is tested by using three scenarios, Scenario 1 using data set from Scholl (1999) with $n = 12$, Scenario 2 using data set from Jackson (1956) in Rahayuningtyas (2011) with $n = 11$, Scenario 3 using data set from Baykasoglu (2006) with $n = 12$. Scenario 1 produces the same solution while scenario 2 and 3 produces a better solution than the research that has been done before.

Keywords: Line Balancing, U-shaped, Guided GRASP, Fixed Threshold.

* 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.

1. PENDAHULUAN

Assembly line balancing (ALB) adalah suatu penugasan penempatan pekerjaan pada stasiun kerja yang saling berhubungan antar lintasan produksi untuk memperoleh target laju produksi yang terpenuhi. Penugasan pekerjaan pada setiap stasiun kerja memiliki ketentuan tidak boleh melebihi dari waktu siklus (*cycle time*) dari stasiun kerja tersebut dan tidak melanggar hubungan yang harus dipenuhi oleh *precedence relationship*.

Permasalahan ALB dapat diklasifikasikan menjadi 4 tipe, *Simple Assembly Line Balancing Problem Type I* (SALBP-I), SALBP II, SALBP-F dan SALBP-E (Scholl dan Becker, 2006). Gutjahr dan Nemhauser (1964) yang terdapat pada Andres *et al* (2003) menggolongkan permasalahan ALB tersebut ke dalam permasalahan optimisasi kombinatorial (NP-Hard). Perubahan jumlah elemen kerja maupun jumlah stasiun kerja yang bertambah besar menyebabkan ruang solusi yang membesar dan membutuhkan waktu komputasi yang lama untuk menyelesaikannya dengan menggunakan metode optimasi. Oleh karena itu dilakukan pendekatan dengan metode heuristik. Metode heuristik dapat menghasilkan solusi dalam waktu relatif singkat meskipun tidak menjamin solusi yang dihasilkan adalah solusi optimal.

Model *U-shaped* muncul karena perkembangan yang dialami pada persoalan area atau luas lahan yang tersedia. Minimisasi lahan terjadi karena bentuk dari lintasan *U-shaped* memiliki rancangan lintasan dengan jalur masuk dengan jalur keluar terletak pada posisi yang sama dan minimisasi jumlah pekerja juga dapat terjadi karena model *U-shaped* memiliki karakteristik operator yang mempunyai kemampuan beragam (*multi skill*). *U-shaped* atau yang biasa dikenal dengan *U-line Assembly Line Balancing Problem* (UALBP) pertama kali dipelajari oleh Miltenburg dan Wijngaard (1994) yang mengajukan program dinamis dan *heuristics single-pass*. Prosedur *U-line Optimizer* (ULINO) diusulkan oleh Scholl dan Klein (1999) dengan tujuan minimisasi jumlah stasiun kerja. Baykasoglu (2006) mengusulkan algoritma *Simulated Annealing* (SA) untuk UALBP dengan kriteria minimisasi jumlah stasiun kerja.

Algoritma *Greedy Randomized Adaptive Search Procedures* (GRASP) adalah salah satu metode *metaheuristic* untuk memecahkan masalah optimisasi kombinatorial. GRASP diusulkan oleh Feo dan Resende (1995) untuk menyelesaikan *traveling salesman problem* dan penjadwalan pengemudi bus. Hidayati (2010) menggunakan algoritma GRASP untuk model keseimbangan lintasan perakitan *U-shaped* dengan kriteria minimisasi jumlah stasiun kerja dan Rahayuningtyas (2011) menggunakan *Guided GRASP* dengan kriteria minimisasi jumlah stasiun kerja. GRASP telah dikembangkan menjadi beberapa jenis yaitu *Multi-Level GRASP*, *Reactive GRASP*, *Guided GRASP*, *Perturbation GRASP*, dan *Paralell*.

Threshold Accepting (TA) dikembangkan oleh Dueck dan Scheuer (1990) untuk menyelesaikan masalah *traveling salesman*. Imran dan Okdinawati (2010) mengembangkan TA untuk permasalahan *Heterogeneous Fleet Vehicle Routing Problem*. TA menerima solusi yang masih dalam ambang batas dari solusi awal untuk dieksplorasi sehingga menemukan solusi yang lebih baik. Fungsi *threshold* tersebut dapat diadaptasikan ke algoritma yang lain. Dari studi literatur, penggabungan algoritma *Guided GRASP* dengan fungsi *threshold* belum dikembangkan sehingga pada penelitian ini dilakukan pengembangan algoritma *Guided GRASP With Fixed Threshold* dengan kriteria minimisasi jumlah stasiun kerja.

Pembahasan yang dilakukan pada penelitian ini adalah tentang keseimbangan lintasan perakitan menggunakan lintasan *U-shaped*. Algoritma yang digunakan untuk pemecahan masalah UALBP yaitu *Guided GRASP With Fixed Threshold* dengan kriteria minimisasi jumlah stasiun kerja dengan maksimisasi efisiensi lintasan perakitan.

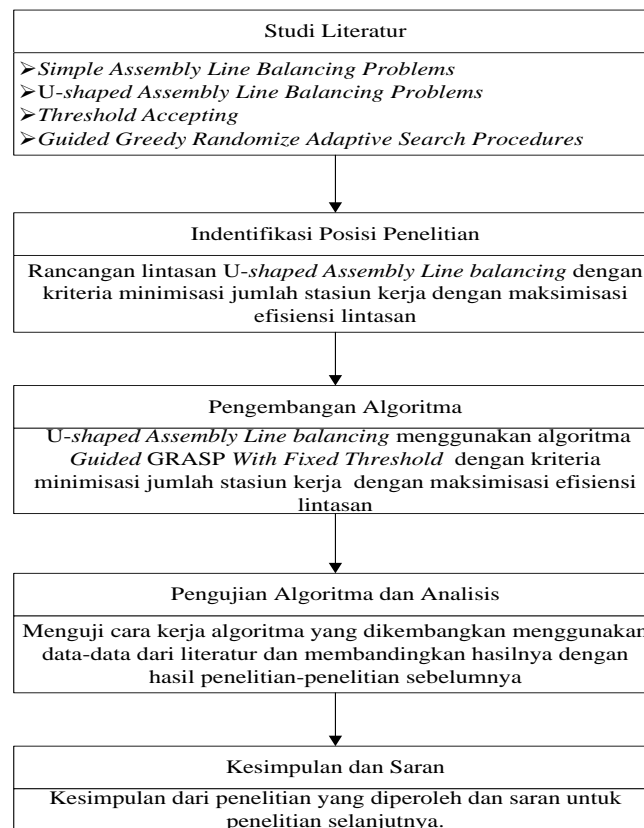
Penelitian ini bertujuan untuk mengembangkan algoritma *Guided GRASP With Fixed Threshold* terhadap lintasan perakitan *U-shaped* dengan kriteria minimisasi jumlah stasiun kerja. Batasan-batasan masalah yang digunakan pada penelitian ini adalah:

1. Sistem yang menjadi pembahasan adalah sistem lintasan perakitan produk tunggal.
2. Mesin atau peralatan yang digunakan untuk merakit produk tidak di bahas.
3. Jumlah operator dalam penelitian ini tidak di bahas.

Beberapa asumsi yang digunakan dalam penelitian adalah: Waktu operasi setiap elemen pekerjaan sudah diketahui secara deterministic, waktu siklus sudah diketahui secara deterministic, waktu *set up* sudah termasuk ke dalam waktu proses dan operator mampu mengerjakan beberapa urutan pekerjaan yang berbeda (*multi skill*).

2. METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan tahap-tahap yang digunakan dalam pengembangan algoritma secara sistematis. Tahapan-tahapan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

Penelitian yang dilakukan dimulai dengan studi literatur terhadap penelitian keseimbangan lintasan lurus (SALBP) dan keseimbangan lintasan *U-shaped* (UALBP) yang diselesaikan dengan algoritma *Guided GRASP With Fixed Threshold*. Studi literaturnya antara lain:

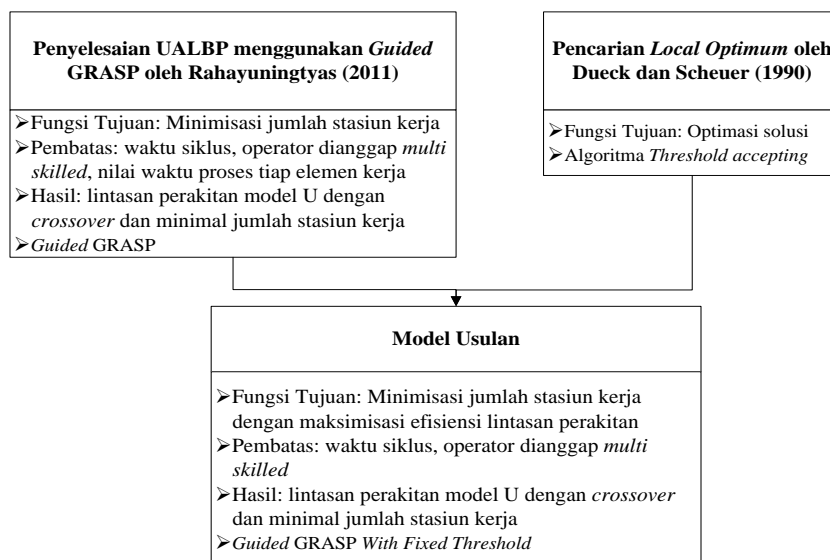
1. Gutjahr and Nemhauser (1964) yang terdapat pada Andres *et al* (2006) menyatakan permasalahan ALB tersebut ke dalam permasalahan optimisasi kombinatorial (NP-*Hard*).
2. Miltenburg dan Wijngaard (1994) melakukan penelitian pertama kali tentang UALBP yang mengusulkan formulasi program dinamis dan aturan prioritas heuristik *single-pass*.

3. Baykasoglu (2006) mengusulkan algoritma *simulated annealing* (SA) untuk menyelesaikan UALBP dengan kriteria minimisasi jumlah stasiun kerja.
4. Hidayati (2010) menyelesaikan UALBP dengan menggunakan algoritma GRASP dengan kriteria minimisasi jumlah stasiun kerja.
5. Rahayuningtyas (2011) menyelesaikan permasalahan U-*Shaped* dengan menggunakan algoritma *Guided* GRASP dengan kriteria minimisasi jumlah stasiun kerja.
6. Dueck dan Scheuer (1990) mengembangkan *Threshold Accepting* (TA) untuk menyelesaikan masalah optimasi kombinatorial pada *travelling salesman problem*.
7. Imran dan Okdinawati (2010) melakukan pengembangan *threshold* untuk menyelesaikan permasalahan *Heterogeneous Fleet Vehicle Routing Problem* (HFVRP).

Berdasarkan studi literatur yang dilakukan, belum ada pengembangan algoritma *Guided GRASP With Fixed Threshold* untuk *U-shaped Assembly Line Balancing Problem* (UALBP). Untuk itu perlu dilakukan penelitian penerapan *Guided GRASP With Fixed Threshold* untuk menyelesaikan permasalahan UALBP dengan kriteria maksimisasi efisiensi lintasan perakitan.

Miltenburg dan Wijngaard (1994) melakukan penelitian pertama kali untuk penyelesaian UALBP yang mengusulkan formulasi program dinamis dan aturan prioritas heuristik single-pass. Hidayati (2010) melakukan penelitian GRASP untuk menyelesaikan UALBP dan Rahayuningtyas (2011) menyelesaikan UALBP menggunakan *Guided GRASP*. Penelitian *Threshold Accepting* dikembangkan oleh Dueck dan Scheuer (1990) untuk pencarian local optimum. Pada penelitian ini penyelesaian UALBP menggunakan algoritma *Guided GRASP With Fixed Threshold* dengan kriteria minimisasi jumlah stasiun kerja dengan maksimisasi efisiensi lintasan perakitan.

Penelitian yang dilakukan merupakan pengembangan algoritma dari penelitian-penelitian sebelumnya seperti yang dilakukan oleh Rahayuningtyas (2011). Skema pengembangan algoritma dapat dilihat pada Gambar 2.



Gambar 2. Skema Pengembangan Algoritma

Pengujian algoritma usulan dilakukan dengan menggunakan tiga skenario, yaitu:

1. Skenario 1 yang bertujuan untuk menguji cara kerja algoritma usulan dengan menggunakan set data Scholl (1999) dengan jumlah elemen pekerjaan (n) = 12.

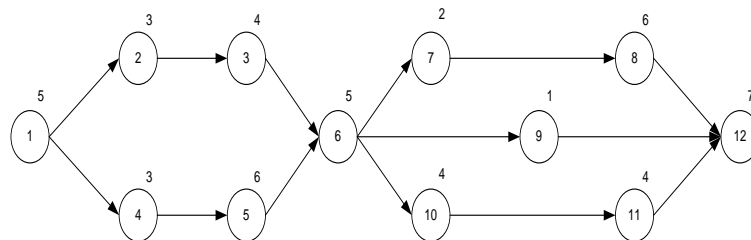
2. Skenario 2 untuk menguji performansi algoritma usulan menggunakan set data Jackson (1956) yang terdapat pada Rahayuningtyas (2011) dengan $n = 11$
3. Skenario 3 untuk menguji cara kerja algoritma usulan menggunakan set data Baykasoglu (2006) dengan $n = 12$.

Hasil pengujian pada kedua skenario tersebut kemudian dibandingkan dengan hasil dari beberapa penelitian yang telah dipublikasikan.

3. PENGEMBANGAN ALGORITMA

3.1 Keseimbangan Lintasan Perakitan U-shaped

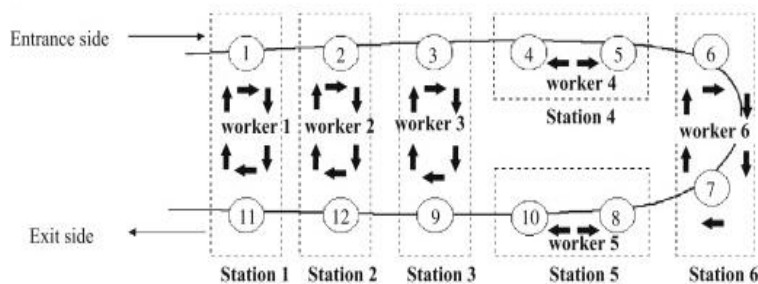
Precedence diagram dari lintasan perakitan tersebut dapat dilihat pada Gambar 3.



Gambar 3. Precedence Diagram

Lingkaran pada *precedencediagram* menunjukkan elemen kerja dan angka yang terdapat didalam lingkaran adalah nomor elemen kerja. angka yang berada diatas lingkaran menunjukkan waktu setiap elemen kerja sedangkan tanda panah yang menghubungkan elemen kerja yang satu dengan yang lainnya disebut *precedence relations*. Pengerjaan setiap elemen kerja harus disesuaikan dengan *precedence relations*.

Sebagai ilustrasi permasalahan pada lintasan perakitan U-shaped dapat dijelaskan sebagai berikut dalam satu lintasan perakitan terdapat 12 elemen kerja yang harus dirakit pada beberapa SK dalam satu lintasan perakitan. Konfigurasi lintasan perakitan U-shaped yang menempatkan 12 elemen kerja ke dalam 6 stasiun kerja ditunjukkan Gambar 4.



Gambar 4. Lintasan perakitan U-Shaped

Semua elemen kerja tersebut dibagi kedalam 6 stasiun kerja (SK1,SK2....SK6) dan masing-masing stasiun kerja terdiri dari 2 elemen kerja. sebagai ilustrasi SK-1 yang terdiri dari 2 elemen kerja yaitu EK1 dan EK11 dikerjakan oleh 1 operator.

3.2 Kerangka Pengembangan Algoritma

Pada sub bab ini akan dibahas mengenai kerangka pengembangan algoritma untuk memecahkan permasalahan U-shaped melalui tahapan algoritma dasar dan algoritma usulan.

3.2.1. Algoritma Dasar

Algoritma dasar dari penelitian mengenai U-shaped pertama kali dipelajari oleh Miltenburg dan Wijngaard (1994) yang menyelesaikan U-shaped Assembly Line balancing Problem (UALBP) menggunakan *Dynamic Programming* (DP) untuk meminimisasi jumlah stasiun kerja. Untuk menyelesaikan *Simple Assembly Line Balancing Problem* (SALBP), Andres *et al* (2003) menggunakan GRASP dengan kriteria minimisasi jumlah stasiun kerja. Selanjutnya Hidayati (2010) menyelesaikan permasalahan U-shaped dengan menggunakan GRASP dan Rahyuningtyas (2011) mengembangkan masalah UALBP menggunakan model *Guided GRASP*.

3.2.2. Algoritma Usulan

Pengembangan model ini dilakukan untuk mendapatkan rancangan lintasan perakitan U-shaped dengan kriteria minimisasi jumlah stasiun kerja dengan maksimisasi efisiensi lintasan. Munculnya lintasan perakitan U-shaped adalah untuk mengatasi kekurangan yang terdapat pada lintasan perakitan lurus. Dasar dari pengembangan algoritma ini adalah penelitian SALBP dengan menggunakan *Greedy Randomized Adaptive Search Procedures* (GRASP) dan penelitian mengenai U-shaped Assembly Line Balancing Problem (UALBP) serta penelitian mengenai UALBP dengan menggunakan GRASP.

Guided GRASP ini akan menghasilkan solusi lebih baik jika ditambahkan fungsi lain, misalnya pada model usulan ini fungsi yang digunakan adalah Δ . Pada *Guided GRASP* ini RCL ditentukan agar menghasilkan elemen-elemen kerja yang akan memberikan solusi lebih baik. oleh karena itu, tidak semua prospektif elemen kerja dapat masuk kedalam RCL dengan kata lain elemen kerja yang terpilih untuk dimasukkan ke dalam stasiun kerja tertentu dengan tidak melebihi waktu siklus (CT) dan tidak melanggar *precedence diagram*.

TA dimulai dari *feasible solution* secara *random*. Kemudian tentukan ambang batas dari *threshold*, ambang batas *threshold* akan menerima solusi yang bukan solusi lebih baik dari solusi inisial tetapi masih dalam rentang ambang batas. Proses pencarian solusi yang terus menerus pada akhirnya akan memperkecil ambang batas *threshold* hingga mencapai nol dan TA akan berubah menjadi pencarian lokal. Fungsi *threshold* tersebut diadopsi ke dalam algoritma *Guided GRASP* untuk memecahkan masalah UALBP. Sehingga penelitian ini menggunakan algoritma *Guided GRASP With Fixed Threshold* untuk menyelesaikan UALBP dengan kriteria minimisasi jumlah stasiun kerja dengan maksimisasi efisiensi lintasan. Dalam penelitian ini persentase *threshold* telah ditentukan sebelumnya yaitu 5%, 15% dan 25%. Persentase *threshold* 5% artinya untuk ambang batas 5% lebih kecil dari solusi inisial yang merupakan solusi yang lebih kecil dari solusi inisial yang masih bisa diterima. Dengan ambang batas tersebut algoritma *Guided GRASP with Fixed Threshold* merupakan algoritma yang mempertimbangkan solusi yang bukan solusi terbaik untuk dilakukan eksplorasi terhadap konfigurasi yang diperoleh.

Notasi-notasi yang digunakan dalam pengembangan algoritma usulan ini adalah sebagai berikut:

- i = Indeks untuk stasiun kerja (SK); ($i = 1, 2, \dots, m$)
- j = Indeks untuk elemen kerja; ($j = 1, 2, \dots, n$)
- t_j = Waktu proses untuk elemen kerja ke- j
- k = Indeks iterasi pada *local search*; ($k = 1, 2, \dots, k_{max}$)
- p = Indeks iterasi keseluruhan ($p = 1, 2, \dots, p_{max}$)
- α = *Threshold* parameter ; $\alpha \in U[0, 5; 1]$
- CT = Waktu siklus
- V = Himpunan elemen kerja yang dapat ditambahkan

- V_c =Himpunan elemen kerja yang dapat dijadikan kandidat prospektif pemilihan RCL
 V_a =Himpunan elemen kerja yang telah ditambahkan ke dalam SK (*assigned*)
 ST_i =Akumulasi waktu elemen kerja pada stasiun kerja i
 EL_k =Efisiensi lintasan pada iterasi k
 SI_k = *Smoothness index* pada iterasi k
 $h(j)$ =Nilai fungsi *greedy* setelah penambahan elemen $j \in RCL$
 $\underline{h}(j)$ =Total waktu SK minimum setelah penambahan elemen kerja j
 $\bar{h}(j)$ =Total waktu SK maksimum setelah penambahan elemen kerja j
 Δ =Selisih waktu siklus dengan akumulasi waktu kerja j pada stasiun kerja i
 EL_p =Efisiensi lintasan pada iterasi p
 EL_{th} =Efisiensi batasan *threshold*
 ($EL_{th} = \% \text{ threshold} \times EL_k$)
 SI_p = *Smoothness index* pada iterasi p

Pengembangan algoritma *Guided GRASP with Fixed Threshold* dengan kriteria minimisasi jumlah stasiun kerja untuk menyelesaikan masalah lintasan perakitan U-Shaped adalah sebagai berikut:

Tahap 1 – *Initial Solution*

Langkah 0

Input $i, j, t_j, V, V_c = \emptyset$, *Precedence diagram*, *Cycle Time*. Set $p_{max} = \max\left(2, \left\lceil \frac{n}{10} \right\rceil\right)$

Langkah 1

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

Langkah 2

1. Perbaharui V_c

V_c adalah elemen kerja dengan *successor* dan *predecessor* terbanyak untuk ditempatkan pada SK i dengan memperhatikan *precedence diagram*.

2. Periksa apakah $|V_c| = 1$?

Jika ya periksa $ST_i > CT$?

Jika ya set $i = i + 1$, jika tidak langsung tempatkan j pada SK i

Periksa apakah $|V_c| = \emptyset$?

Jika ya, berhenti dan lanjutkan ke Tahap 2– *Local Search*

Jika tidak kembali ke Langkah 2

Jika tidak kembali ke Langkah 3

Langkah 3

1. Tentukan ST_i minimum dan ST_i maksimum

2. Bangkitkan nilai α dengan rentang nilai 0,5-1 secara *random* $\rightarrow \alpha \in U [0,5;1]$

3. Bentuk RCL

4. Tentukan L_{min}

5. Periksa apakah $|RCL| < L_{min}$? Jika ya lakukan *relaxing*

Urutkan elemen kerja $j \in V_c$ dari t_j terkecil ke terbesar, pilih urutan sebanyak L_{min} untuk masuk ke dalam RCL dan lanjutkan ke Langkah 4.

Jika tidak lanjutkan ke Langkah 4.

Langkah 4

Hitung delta (Δ) = $CT - ST_i$ untuk $j \in RCL$ kemudian pilih elemen kerja j dengan nilai delta (Δ) terkecil untuk ditempatkan pada SK i dengan waktu stasiun tidak melebihi CT dan tidak melanggar *precedence diagram*. Lanjutkan ke Langkah 5.

Langkah 5

Periksa apakah $ST_i > CT$?

Jika ya, set $i = i + 1$; tempatkan j kedalam V_a pada SK i

Jika tidak, langsung tempatkan j ke dalam V_a pada SK i lanjutkan ke Langkah 6.

Langkah 6

Periksa apakah $|V| = \emptyset$

Jika ya, maka solusi telah terbentuk karena tidak ada lagi elemen kerja yang harus ditempatkan pada stasiun kerja dan lanjutkan ke Tahap 2 – *Local Search*.

Tahap 2 *Local Search*

Langkah 7

Input data konfigurasi Tahap 1 dan % *threshold*

Langkah 8

Set $k = 1$. Set $k_{max} = \max\left(4, \left\lceil \frac{n}{4} \right\rceil\right)$. Set $EL = EL_0$

Langkah 9

Lakukan *crossover* pada lintasan tersebut dengan cara menukar elemen kerja ke- j ataupun menambahkan elemen kerja ke- j dengan tetap tidak melebihi CT , tidak melanggar *precedence constraints*, dan lintasan tetap berbentuk U.

1 Lakukan proses *exchange*

1.1 Ambil salah satu elemen kerja ke- j secara *random*.

1.2 Lakukan proses *exchange* untuk elemen kerja j yang terpilih dengan elemen kerja j lainnya.

1.3 Apakah proses *exchange* melanggar *precedence constraints*?

Jika ya lanjutkan ke Langkah 1.4, jika tidak lanjutkan ke Langkah 1.5

1.4 Apakah masih terdapat j yang belum ditukar?

Jika ya kembali ke Langkah 1.2, jika tidak lanjut ke Langkah 2.9

1.5 Apakah masih ada j yang dapat ditukar?

Jika ya kembali ke Langkah 1.2, jika tidak lanjutkan ke Langkah 1.6

1.6 Hitung nilai EL dan SI . Apakah $EL_{th} \leq EL_k \leq 100\%$?

Jika ya lanjutkan ke Langkah 1.7, jika tidak kembali ke Langkah 1.4

1.7 Apakah $|EL_{th} \leq EL_k \leq 100| > 1$

Jika ya pilih j dengan nilai EL terbesar, jika terdapat EL yang sama pilih nilai SI terkecil, jika terdapat nilai SI yang sama pilih secara *random* dan lanjut ke Langkah 1.8, jika tidak lanjut ke Langkah 1.8

1.8 Apakah $EL = 100\%$?

Jika ya lanjut ke Langkah 12, jika tidak lanjut ke Langkah 2.9

2 Lakukan proses *insert*

2.1 Pilih salah satu SK ke- i secara *random*.

2.2 Lakukan proses *insert* untuk SK i yang terpilih dengan elemen kerja j lainnya.

2.3 Apakah proses *insert* melanggar *precedence constraints*

Jika ya lanjutkan ke Langkah 2.4, jika tidak lanjutkan ke Langkah 2.5

2.4 Apakah masih terdapat j yang belum di *insert*?

Jika ya kembali ke Langkah 2.2, jika tidak lanjut ke Langkah 2.7

2.5 Apakah masih ada j yang dapat di *insert*?

Jika ya kembali ke Langkah 2.2, jika tidak lanjutkan ke Langkah 2.6

2.6 Apakah terdapat solusi *insert* yang menyebabkan SK berkurang?

Jika ya periksa apakah $EL_k = 100\%$?

Jika ya lanjutkan ke Langkah 12

Jika tidak maka Lanjutkan ke Langkah 2.13.

Jika tidak lanjutkan ke Langkah 2.7.

2.7 Hitung nilai EL dan SI . Apakah $EL_{th} \leq EL_k \leq 100\%$?

Jika ya lanjutkan ke Langkah 2.8, jika tidak kembali ke Langkah 2.4

2.8 Apakah $|EL_{th} \leq EL_k \leq 100\%| > 1$

Jika ya pilih j dengan nilai EL terbesar, jika terdapat EL yang sama pilih nilai SI terkecil, jika terdapat nilai SI yang sama pilih secara *random* dan lanjut ke Langkah 2.9, jika tidak lanjut ke Langkah 2.9.

2.9 Apakah $EL = 100\%$?

Jika ya lanjut ke Langkah 12, jika tidak lanjutkan ke Langkah 2.10.

2.10 Apakah $EL_k \geq EL_{k-1}$

Jika ya set $EL = EL_k$ dan lanjutkan ke Langkah 2.11.

Jika tidak lanjutkan ke Langkah 2.12

2.11 Gunakan konfigurasi terbaru dan lanjutkan ke Langkah 2.13.

2.12 Gunakan konfigurasi sebelumnya dan lanjutkan ke Langkah 2.13.

2.13 Set $k = k + 1$, periksa apakah $k > k_{max}$?

Jika ya lanjut ke Langkah 11, jika tidak kembali ke Langkah 9

Langkah 10

Pilih nilai EL terbaik sebagai inisial solusi iterasi selanjutnya dan lanjutkan ke Langkah 11

Langkah 11

Set $p = p + 1$, Periksa apakah $p \geq p_{max}$?

Jika ya maka pilih *final local optima* dengan EL terbesar atau *SI* terkecil dan berhenti dan lanjutkan ke Langkah 12, jika tidak kembali ke Langkah 2

Langkah 12

Tampilkan konfigurasi terbaik

3.2.3. Perbedaan Algoritma Usulan Dengan Algoritma Guided Grasp

Perbedaan algoritma usulan dengan algoritma Guided GRASP terdapat pada bagian local search. Pada tahap *local search* yang ditambahkan dengan fungsi *threshold*. Fungsi *threshold* ini untuk penentuan solusi yang dicari untuk menghasilkan solusi terbaik. Fungsi *threshold* ini dapat mengakomodasi solusi yang bukan solusi terbaik tetapi masih dalam batas toleransi untuk diteruskan ke tahap selanjutnya. Fungsi *threshold* ini menyebabkan ruang untuk pencarian solusi menjadi lebih luas. Proses *threshold* terjadi pada proses *exchange* dan *insert*, yaitu elemen kerja yang dapat ditukar kemudian diperiksa efisiensi lintasannya dan dibandingkan dengan nilai *threshold* yang ditetapkan, jika elemen kerja tersebut dapat di *exchange* dan solusi yang dihasilkan masih dalam *threshold*, maka proses dapat dilanjutkan ke tahap *insert*. Pada tahap *insert* jika terdapat elemen kerja yang dapat di *insert* dan solusi yang dihasilkan masih dalam ambang *threshold* maka solusi tersebut dapat dilakukan perbandingan dengan iterasi selanjutnya.

4. PENGUJIAN ALGORITMA DAN ANALISIS

4.1 Pengujian Algoritma

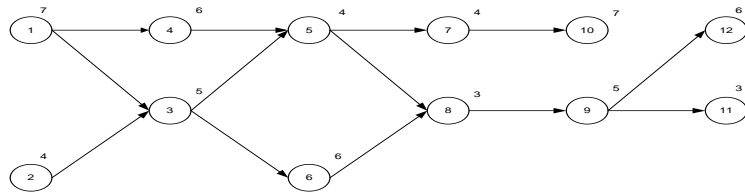
Pengujian algoritma Guided GRASP with Fixed Threshold terdiri dari Skenario 1 dan Skenario 2. Skenario 1 yang digunakan untuk menguji cara kerja dari pengembangan algoritma usulan adalah set data Scholl (1999), sedangkan untuk Skenario 2 yang digunakan untuk menguji pengembangan algoritma usulan yaitu set data Jackson (1956) yang terdapat pada Rahayuningtyas (2011) dan Baykasoğlu (2006). Setiap skenario diuji dengan 3 nilai *threshold* yang telah di tentukan yaitu 5%, 15%, dan 25%.

Set data yang digunakan pada tahap pengujian algoritma yaitu:

1. Set data Scholl (1999), memiliki waktu siklus (CT) = 10. Set data berupa elemen kerja (j) dan waktu elemen kerja (t_j) dapat dilihat pada Tabel 1 dan *precedence diagram* pada Gambar 5.

Tabel 1. Set Data Scholl (1999)

j	1	2	3	4	5	6	7	8	9	10	11	12
t_j	7	4	5	6	4	6	4	3	5	7	3	6
Successor	4,3	3	5,6	5	7,8	8	1	9	11,12			

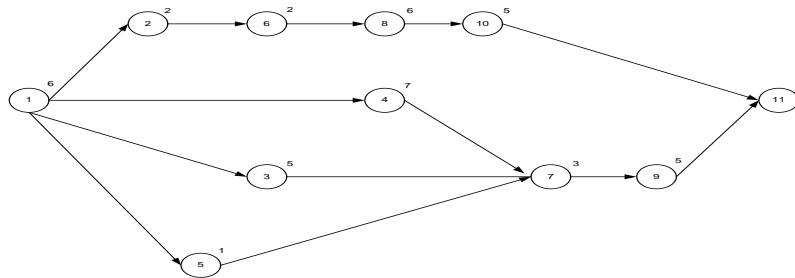


Gambar 5. Precedence Diagram Scholl (1999)

2. Set data Jackson (1956) dalam Rahayuningtyas (2011) memiliki jumlah elemen kerja (n) = 11 dengan waktu siklus (CT) = 10. Data elemen kerja (j) dan waktu elemen kerja (t_j) serta *precedence diagram* dapat dilihat pada Tabel 2 dan Gambar 6.

Tabel 2. Set Data Jackson (1956)

j	1	2	3	4	5	6	7	8	9	10	11
t_j	6	2	5	7	1	2	3	6	5	5	4
Successor	2,3,4,5	6	7	7	7	8	9	10	11	11	

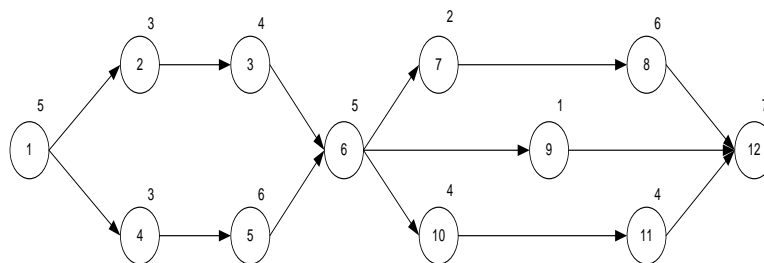


Gambar 6. Precedence Diagram Jackson (1956)

3. Set data Baykasoglu (2006), memiliki jumlah elemen kerja (n) = 12 dengan waktu siklus (CT) = 10. Set data berupa elemen kerja (j) dan waktu elemen kerja (t_j) dapat dilihat pada Tabel 3 dan *precedence diagram* pada Gambar 7.

Tabel 3. Set Data Baykasoglu (2006)

j	1	2	3	4	5	6	7	8	9	10	11	12
t_j	7	4	5	6	4	6	4	3	5	7	3	6
Successor	2,4	3	6	5	6	7,9,10	8	12	12	11	12	



Gambar 7. Precedence Diagram Baykasoglu (2006)

Perbandingan hasil algoritma *Guided GRASP with Fixed Threshold* dan perbandingan hasil algoritma *Guided GRASP with Fixed Threshold* dengan penelitian sebelumnya dapat dilihat pada Tabel 4 dan 5.

Tabel 4. Perbandingan Hasil Algoritma *Guided GRASP with Fixed Threshold*

Set Data	% <i>threshold</i>	Jumlah elemen Kerja (<i>n</i>)	Waktu Siklus (<i>CT</i>)	Jumlah Stasiun Kerja	EL %	<i>SI</i>
Jackson (1956)	5	11	10	5	92	2,449
	15		10	5	92	2,449
	25		10	5	92	2,449
Scholl (1999)	5	12	10	6	100	0
	15		10	6	100	0
	25		10	6	100	0
Baykasoğlu (2006)	5	12	9	6	92,593	2
	15		9	6	92,593	2
	25		9	6	92,593	2

Tabel 5. Perbandingan Hasil Algoritma *Guided GRASP with Fixed Threshold* dengan Penelitian Sebelumnya

Set Data	% <i>threshold</i>	Jumlah elemen Kerja (<i>n</i>)	Waktu Siklus (<i>CT</i>)	Jumlah Stasiun Kerja	EL %	<i>SI</i>
Jackson (1956)	5	11	10	5	92	2,449
Scholl (1999)			10	7	85,71	5,381
Baykasoğlu (2006)			9	6	92,593	2
Jackson (1956)	15	12	10	5	92	2,449
Scholl (1999)			10	6	100	0
Baykasoğlu (2006)			9	6	92,593	2
Jackson (1956)	25	12	10	5	92	2
Scholl (1999)			10	6	100	0
Baykasoğlu (2006)			9	6	92,593	2

4.2 Analisis

Berdasarkan hasil yang diperoleh dari algoritma *Guided GRASP with Fixed Threshold* pada Tabel 4, nilai *threshold* 15%, 25% dan 5% menghasilkan jumlah stasiun kerja, nilai EL dan *SI* yang sama. Nilai *threshold* 5% berarti ambang batas untuk menerima solusi yang lebih jelek yang diperbolehkan semakin kecil, sehingga ruang solusi untuk proses *local search* menjadi lebih sedikit. Nilai *threshold* 15% dan 25% yang memiliki ambang batas toleransi yang lebih besar. Sehingga ruang solusi dari proses *local search* menjadi lebih banyak.

Dari Tabel 5 untuk set data Baykasoglu (2006) dapat dikatakan algoritma usulan lebih baik. Hal ini dapat dilihat pada perbandingan nilai *EL* dan *SI* untuk algoritma usulan lebih besar yaitu 92,59% daripada penelitian Rahayuningtyas (2011) dan Baykasoglu (2006) yaitu 83,33%. Untuk nilai *SI* algoritma usulan juga memberikan hasil lebih kecil yaitu 2, sedangkan Rahayuningtyas (2011) dan Baykasoglu (2006) berturut-turut 10,48 dan 5,09. Untuk set data Jackson (1956) dan Scholl (1999) baik algoritma usulan maupun penelitian dari Rahayuningtyas (2011), Scholl dan Klein (1999) serta Miltenburg dan Wijngaard (1994) memiliki nilai EL yang sama yaitu 92%. Tetapi untuk set data Jackson (1956) nilai *SI* dari algoritma usulan lebih kecil daripada penelitian sebelumnya oleh Rahayuningtyas (2011) serta Miltenburg dan Wijngaard (1994) masing-masing yaitu 4 dan 2,45.

Algoritma usulan memiliki proses *local search* dimulai dengan *exchange* dan *insert*. Jika terdapat elemen kerja yang dapat di *exchange* maka langsung dilakukan proses *insert* dengan mempertimbangkan nilai *threshold*. Hal ini menyebabkan keefektifan dari fungsi *local*

search dapat tercapai karena untuk setiap *exchange* yang diijinkan langsung dilakukan proses *insert* untuk mencari solusi yang lebih baik. Fungsi *threshold* juga membantu untuk proses *local search* yang solusi inisial lebih kecil daripada solusi awal tetapi masih di ambang batas toleransi dan ada kemungkinan solusi inisial tersebut lebih baik daripada solusi awal. Hal ini menunjukkan algoritma usulan dapat memecahkan permasalahan UALBP dengan hasil yang baik.

5. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian pengembangan algoritma *Guided GRASP with Fixed Threshold* adalah sebagai berikut:

1. Penelitian ini menghasilkan algoritma *Guided GRASP with Fixed Threshold* untuk model lintasan perakitan *U-shaped* dengan ukuran performansi minimisasi jumlah stasiun kerja.
2. Penelitian ini dikembangkan dari penelitian Rahayuningtyas (2011) yang menggunakan algoritma *Guided GRASP*.
3. Pengujian data untuk Skenario 1 menunjukkan algoritma usulan dapat diaplikasikan pada set data Scholl (1999) untuk menyelesaikan UALBP dengan kriteria maksimisasi efisiensi lintasan perakitan.
4. Pengujian set data untuk Skenario 2 menunjukkan algoritma usulan dapat mengeksplorasi solusi inisial dari set data Jackson (1956) dan Baykasoglu (2006) menjadi lebih baik. Untuk set data Jackson (1956) yang terdapat pada Rahayuningtyas (2011) menghasilkan solusi inisial yang sama sedangkan Baykasoglu (2006) menghasilkan EL yang lebih baik. Untuk dipertimbangkan pada penelitian selanjutnya adalah dengan melakukan pengembangan model *U-Shaped* dengan menggunakan metode *Threshold Accepting*.

DAFTAR PUSTAKA

- Andres, G. R., Schniederjans, M. J., dan Olson, J. R. (2003), U-OPT: An Analysis of Exact U-shaped Line balancing Procedures. *International Journal of Production Research*, 41,4185-4210.
- Baykasoglu, A. (2006), Multi-rule Multi-objective Simulated Annealing Algorithm for Straight an U-type Assembly Line Balancing Problems. *Journal of Intelligent Manufacturing*, 17 215-232.
- Dueck, G. dan Scheuer, T. (1990). Threshold Accepting. A General Purpose Optimization Algorithm Superior to Simulated Annealing. *Journal of Computational Physics* 90, 161-175.
- Feo, T. A., Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6,109-133.
- Hidayati, N., 2010, Model Keseimbangan Lintasan Perakitan *U-Shaped* Menggunakan *Greedy Randomized Adaptive Search Procedures* dengan Kriteria Minimisasi Jumlah Stasiun Kerja.
- Imran, A., dan Okdinawati, L. (2010). A Threshold Accepting Heuristic For The Heterogeneous Fleet Vehicle Routing Problem. *Jurnal Logistik Bisnis*.
- Miltenburg, J., dan Wijngaard, J. (1994), The U-line Balancing Problem. *Management Science*, 40, 1378-1388.
- Rahayuningtyas, Y. (2011), Model Keseimbangan Lintasan Perakitan *U-Shaped* Menggunakan *Guided Greedy Randomized Adaptive Search Procedures* dengan Kriteria Minimisasi Jumlah Stasiun Kerja.
- Scholl, A. (1999), *Balancing and Sequencing of Assembly Lines*. Second Edition. Physica Verlag Heidelberg New York.
- Scholl, A., & Becker, C. (2006). State of The Art Exact and heuristic Solution Procedures for Simple Assembly Line Balancing. *European Journal of Operational Research*, 168,666-693.