

# Algoritma Penjadwalan *Job Shop* Kelompok Mesin Homogen dan Heterogen Menggunakan *Variable Neighborhood Descent with Fixed Threshold* Menggunakan Kriteria Minimasi *Makespan*\*

MALINDA MULYAWATI UTOMO, ARIF IMRAN, EMSOSFI ZAINI

Jurusan Teknik Industri  
Institut Teknologi Nasional (Itenas) Bandung

Email: malinda.utomo@gmail.com

## ABSTRAK

*Makalah ini membahas penjadwalan job shop kelompok mesin homogen dan heterogen menggunakan algoritma variable neighborhood descent (VND) with fixed threshold dengan kriteria minimasi makespan. Tahap-tahap yang dilakukan dalam algoritma ini yaitu tahap konstruksi yang diperbaiki dengan menggunakan local search. Pada tahap konstruksi, inisial solusi dibangkitkan dengan penjadwalan non delay, setelah itu jadwal tersebut diperbaiki pada tahap local search dimana local search yang digunakan adalah exchange dan insert terhadap struktur neighborhood secara deterministik. Data yang digunakan pada penelitian ini merupakan data-data dari literatur. Hasil yang diperoleh dari penelitian ini sama dengan penelitian sebelumnya yang dilakukan oleh Putra (2010).*

**Kata kunci:** *Job shop, kelompok mesin, variable neighborhood descent with fixed threshold*

## ABSTRACT

*This paper discusses the scheduling of job shop with machine groups homogen and heterogen using variable neighborhood descent (VND) with fixed threshold for the makespan minimization criteria. The stage are performed on this algorithm, namely the construction phase is improved by using of local search. At this stage of constructs, the initial scheduling solution generated by the non-delay, after which the schedule is fixed at a stage where local search local search is used to exchange and insert process of deterministic structure in neighborhood. The data used in this paper is the data form the literature. The results obtained from this study together with the previous intensive search conducted by Putra (2010).*

**Keywords:** *Job shop, machine groups, variable neighborhood descent with 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

### 1.1 Latar Belakang

Permasalahan penjadwalan *job shop* klasik yaitu ada sejumlah  $n$  *job* pada  $m$  mesin dengan setiap *job* dapat memiliki urutan mesin yang berbeda dengan *job* lainnya, tujuan dilakukannya penjadwalan adalah untuk menghasilkan jadwal yang dapat meminimumkan waktu yang dibutuhkan untuk memproses seluruh *job* pada semua mesin (*makespan*). Dalam industri manufaktur, stasiun kerja yang digunakan yaitu mesin tunggal dan mesin paralel. Beberapa jenis mesin paralel dalam satu stasiun kerja terdiri dari kelompok mesin yang dapat memiliki karakteristik sama (homogen) atau berbeda (heterogen).

Beberapa penelitian yang telah dikembangkan sebelumnya mengasumsikan bahwa setiap stasiun kerja hanya dapat memproses satu *job* pada saat bersamaan, karena di setiap stasiun kerja hanya terdapat satu mesin. Pada kenyataannya banyak dijumpai stasiun kerja dengan jumlah mesin lebih dari satu atau bisa disebut dengan kelompok mesin. Beberapa penelitian yang membahas penjadwalan *job shop* kelompok mesin paralel homogen dan heterogen dengan kriteria minimasi *makespan* diantaranya, Puryani (2003) membahas penjadwalan *job shop* kelompok mesin paralel menggunakan Algoritma Jadwal Aktif dan Algoritma *Branch and Bound Like* (BABL). Luis et al. (2004) mengajukan Algoritma *Ant Colony System* (ACS) untuk menyelesaikan masalah penjadwalan *job shop* kelompok mesin paralel homogen. Putra (2010) membahas permasalahan *job shop* kelompok mesin paralel untuk mesin heterogen dan homogen dengan menggunakan algoritma *Greedy Randomized Adaptive Search Procedure* (GRASP), maka diperlukan pengembangan suatu model penjadwalan *job shop* kelompok mesin paralel menggunakan *Variable Neighborhood Descent with Fixed Threshold* dengan kriteria minimasi *makespan* untuk memecahkan masalah tersebut.

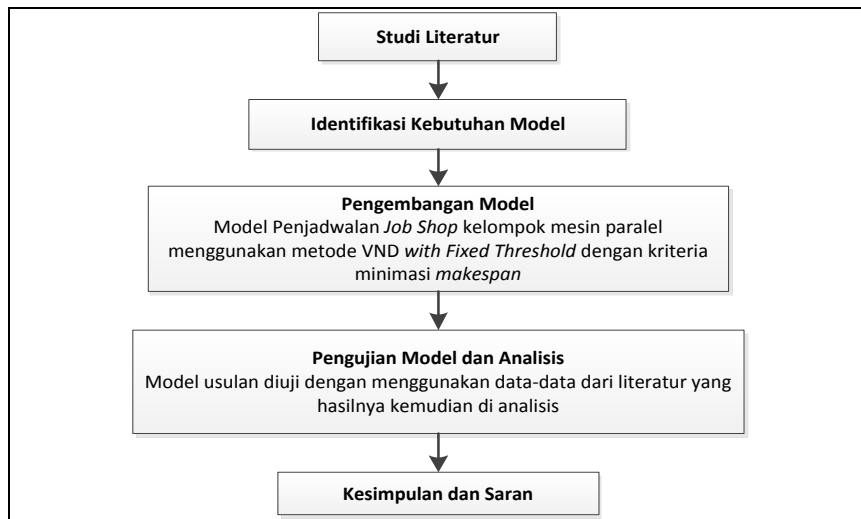
*Variable Neighbourhood Descent* (VND) merupakan metode metaheuristik yang dapat menyelesaikan masalah-masalah *combinatorial optimization* dengan cara menggunakan *neighbourhood* yang berbeda-beda secara terstruktur (Mladenovic dan Hensen, 1997). Pada VND perubahan *neighbourhood* dilakukan secara deterministik. *Threshold accepting* adalah sebuah metode metaheuristik untuk pencarian solusi optimum melalui bilangan *random feasible* dengan mengubah set *neighborhood*. Algoritma *threshold accepting* memungkinkan solusi tidak terjebak pada minimum lokal karena menerima solusi baru yang mengarah ke nilai yang lebih tinggi (Dueck dan Scheuer (1990).

### 1.2 Identifikasi Masalah

Pada kondisi nyata dalam industri manufaktur, stasiun kerja dapat terdiri atas beberapa mesin paralel. Penelitian ini akan dibahas sistem produksi *job shop* dengan kelompok mesin paralel. Sistem ini dapat dijelaskan sebagai sejumlah *job* harus dikerjakan pada satu set stasiun kerja dengan setiap stasiun kerja dapat terdiri dari beberapa mesin paralel. Beberapa mesin paralel dalam satu stasiun kerja ini disebut sebagai kelompok mesin yang dapat memiliki karakteristik sama (homogen) atau berbeda (heterogen). Algoritma yang digunakan adalah VND *with Fixed Threshold* dan kriteria performansi yang digunakan adalah minimisasi *makespan*.

## 2. METODOLOGI PENELITIAN

Langkah-langkah penelitian ini dapat dilihat pada Gambar 1.



**Gambar 1. Langkah-langkah Penelitian**

Peta posisi penelitian terhadap beberapa penelitian yang lalu dapat dilihat pada Gambar 2.

			Heuristic	Metaheuristic			
				Ant Colony System	GRASP	VND	VND with FixedThreshold
Job Shop	Kelompok Mesin Paralel	Homogen	Puryani (2003)	Luis et al. (2003)	Putra (2010)		Penelitian
		Heterogen	Puryani (2003)				

**Gambar 2. Peta Posisi Penelitian**

### 3. PENGEMBANGAN ALGORITMA

Algoritma yang dikembangkan dalam penelitian ini merupakan pengembangan dari penelitian-penelitian sebelumnya, yaitu:

1. Puryani (2003) membahas penjadwalan *job shop* kelompok mesin paralel untuk mesin homogen dan heterogen dengan menggunakan algoritma *Branch and Bound Like (BABL)* untuk minimasi *makespan*.
2. Putra (2010) membahas permasalahan *job shop* kelompok mesin paralel untuk mesin heterogen dan homogen dengan menggunakan algoritma *Greedy Randomized Adaptive Search Procedure (GRASP)* dengan kriteria *makespan*.

### 4. PENGUJIAN ALGORITMA DAN ANALISIS

#### 4.1 Pengujian Algoritma

Tahap 1 menggunakan set data dari Puryani (2003) kasus 1.

Tahap 2 menggunakan set data dari Puryani (2003) kasus 2 dan kasus 3.

*Routing* stasiun kerja yang digunakan untuk semua kasus dapat dilihat pada Tabel 1. Waktu proses mesin paralel homogen dan heterogen dapat dilihat pada Tabel 2 sampai dengan Tabel 7.

**Tabel 1. Routing Stasiun Kerja dari Masing-masing Job untuk Mesin Paralel Homogen dan Heterogen.**

Jumlah Job	Job Ke-	Operasi Ke-		
		1	2	3
5	A	1	2	3
	B	2	1	3
	C	1	2	3
	D	2	1	3
	E	2	1	3

**Tabel 2. Waktu Proses Kasus 1 Mesin Paralel Homogen**

Job Ke-	Operasi Ke-					
	1		2		3	
	m1	m2	m1	m2	m1	m2
A	4	4	8	8	4	4
B	5	5	6	6	2	2
C	2	2	6	6	9	9
D	4	4	9	9	2	2
E	6	6	7	7	2	2

**Tabel 3. Waktu Proses Kasus 1 Mesin Paralel Heterogen**

Job Ke-	Operasi Ke-					
	1		2		3	
	m1	m2	m1	m2	m1	m2
A	4	4	10	5	7	1
B	9	1	5	6	2	2
C	1	3	8	4	9	8
D	3	5	8	9	2	1
E	6	6	10	3	6	4

**Tabel 4. Waktu Proses Kasus 2 Mesin Paralel Homogen**

Job Ke-	Operasi Ke-						
	1		2		3		
	m1	m2	m1	m2	m1	m2	m3
A	5	5	2	2	6	6	6
B	2	2	4	4	6	6	6
C	3	3	5	5	6	6	6
D	3	3	8	8	7	7	7
E	5	5	7	7	8	8	8

**Tabel 5. Waktu Proses Kasus 3 Mesin Paralel Heterogen**

Job Ke-	Operasi Ke-						
	1		2		3		
	m1	m2	m1	m2	m1	m2	m3
A	8	4	2	2	8	6	4
B	1	2	1	6	4	9	3
C	6	3	3	7	10	6	3
D	3	2	9	6	9	8	4
E	7	2	4	9	9	6	8

**Tabel 6. Waktu Proses Kasus 3 Mesin Paralel Homogen**

Job Ke-	Operasi Ke-								
	1			2			3		
	m1	m2	m3	m1	m2	m3	m1	m2	m3
A	6	6	-	7	7	7	6	6	6
B	8	8	8	5	5	-	3	3	3
C	3	3	-	3	3	3	4	4	4
D	7	7	7	4	4	-	3	3	3
E	3	3	3	8	8	-	6	6	6

**Tabel 7. Waktu Proses Kasus 3 Mesin Paralel Heterogen**

Job Ke-	Operasi Ke-								
	1			2			3		
	m1	m2	m3	m1	m2	m3	m1	m2	m3
A	1	10	-	1	10	9	2	6	9
B	7	5	10	6	3	-	4	1	2
C	2	3	-	1	5	2	1	8	2
D	2	9	10	3	5	-	1	1	7
E	4	3	2	6	10	-	10	7	1

Notasi-notasi yang digunakan dalam algoritma penjadwalan adalah:

- $n$  = jumlah *job*.
- $m$  = jumlah mesin.
- $i$  = indeks *job* (1, 2, ..., n).
- $j$  = indeks operasi *job*  $i$ .
- $k$  = indeks stasiun kerja (1, 2, ..., k).
- $r$   $ij$  = *ready time job*  $i$  operasi  $j$  siap untuk dijadwalkan.
- $R_{km}$  = *ready time* stasiun kerja  $k$  dan mesin  $m$ .
- $km^*$  = stasiun kerja  $k$  dan mesin  $m$  yang terpilih.
- $r$  = *stage* jadwal ke  $r$ .
- $PS_r$  = jadwal parsial.
- $S_r$  = set operasi yang dapat dijadwalkan pada *stage*  $r$ , setelah diperoleh  $PS_r$ .
- $\sigma_{ijkm}$  = waktu tercepat operasi  $j$   $S_r$  dapat dimulai.
- $\sigma_{ij}^*$  = waktu tercepat operasi  $j$   $S_r$  yang terpilih dari  $\sigma_{ijkm}$ .
- $\sigma^*$  = waktu tercepat operasi  $j$   $S_r$  yang terpilih dari  $\sigma_{ij}^*$ .
- $C_{ijkm}$  = waktu selesai *job*  $i$  operasi  $j$  di stasiun kerja  $k$  dan mesin  $m$ .
- $C_{ijkm}(iter)$  = waktu selesai *job*  $i$  operasi  $j$  di stasiun kerja  $k$  dan mesin  $m$  dalam iterasi.
- $t_{ijkm}$  = waktu proses *job*  $i$  operasi  $j$  di stasiun kerja  $k$  dan mesin  $m$ .
- $Iter$  = iterasi ke-.
- $MS$  = *makespan*.
- $MS_{iter}$  = *makespan* dalam iterasi.
- $MS_{th}$  = *makespan threshold*.
- $\%Th$  = persentase batas *threshold*.
- $S$  = jadwal terbaik.
- $S'$  = jadwal terbaik setelah dilakukan *exchange* dan *insert*.
- $LK$  = lintasan kritis yang terbentuk.
- $W$  = himpunan pasangan operasi yang dipertukarkan.
- $X$  = jumlah pasangan operasi yang dipertukarkan.
- $Y$  = pasangan operasi ke  $Y$ ,  $Y = (1, 2, \dots, X)$  dalam  $W$  yang akan dipertukarkan.
- $Z$  = himpunan operasi-operasi *job* terakhir yang berada pada lintasan kritis
- $Ob$  = Operasi terakhir dalam lintasan kritis.

$\Omega_{PM}[h]$  = himpunan operasi-operasi yang mendahului operasi  $h$  untuk mesin paralel yang sama.

$\Omega_{PJ}[h]$  = himpunan operasi-operasi yang mendahului operasi  $h$  untuk  $job$  yang sama.

Tahap-tahap yang dilakukan pada algoritma usulan adalah sebagai berikut:

1. Tahap Konstruksi  
Tahap konstruksi adalah tahap pembentukan inisial solusi dengan menggunakan metode penjadwalan *non delay*.
2. Tahap Local Search  
Tahap ini adalah tahap pencarian solusi lebih baik dengan cara merubah struktur *neighborhood*. Terdapat 2 struktur *neighborhood* dalam algoritma usulan ini:
  - a. *Insert*  
Proses *insert* merupakan perubahan struktur dengan cara menyisipkan sebuah operasi kedalam operasi lainnya.
  - b. *Exchange*  
Proses *exchange* merupakan perubahan struktur dengan cara melakukan pertukaran posisi dari dua operasi terpilih pada satu stasiun kerja.

**Tahap Konstruksi:** penjadwalan dibuat menggunakan aturan penjadwalan *non delay*.

#### Langkah 1

*Input data:* matriks *routing data* dan matriks waktu proses setiap *job*.

Set: - *Ready time* operasi pertama dari setiap *job* sama dengan nol.

$$[r_{ij} = 0, \quad i = (1, 2, \dots, n)]$$

- *Ready time* seluruh mesin sama dengan nol.

$$[R_{km} = 0, \quad k = (1, 2, \dots, k) \quad m = (1, 2, \dots, m)]$$

- Tentukan  $r = 0$

#### Langkah 2

Mulai dengan  $P_{Sr}$  sebagai jadwal parsial. Tentukan seluruh operasi tanpa *predecessor* untuk semua alternatif kelompok mesin paralel homogen dan heterogen sebagai  $Sr$ .

#### Langkah 3

Tentukan  $\sigma_{ij}^* = \min \{ \sigma_{ijkm} \}, j \in Sr$  dan tentukan  $\sigma^* = \min \{ \sigma_{ij}^* \}, j \in Sr$  untuk mengetahui *job* dan operasi yang dapat direalisasikan. Setelah itu tentukan  $km^*$  yaitu stasiun kerja dan mesin tempat  $\sigma^*$  dapat direalisasikan.

#### Langkah 4

Jadwalkan *job* yang memiliki  $\sigma_{ij} = \sigma_{ij}^*$  dengan  $km^*$  yang akan direalisasikan pada  $P_{Sr}$ , jika terdapat beberapa *job* yang memiliki  $\sigma^*$  sama, maka pilih salah satu *job* pada tiap stasiun kerja yang berbeda dengan mesin yang berbeda untuk dijadwalkan dengan kriteria saat selesai tercepat ( $C_{ijkm}$ ).

#### Langkah 5

Untuk setiap jadwal parsial baru yang dihasilkan pada langkah 5, perbaharui set data berikut:

- Keluarkan operasi  $j$  dari  $Sr$  termasuk operasi pada kelompok mesin paralel.
- Tambahkan operasi berikutnya dari *job* yang sama ke dalam  $Sr$ .
- Set  $r = r + 1$ .

#### Langkah 6

Periksa apakah  $Sr \neq \{ \}$ .

Jika ya, kembali ke langkah 3 sampai suatu jadwal *non delay* dihasilkan.  
Jika tidak, lanjutkan ke langkah 8.

#### Langkah 7

Tampilkan *Gantt Chart* hasil penjadwalan, dan jadikan sebagai solusi inisial.  
hitung *completion time* dan *makespan* dari *job* yang akan direalisasikan.

$$C_{ijkm} = [r_{ij}, R_{km}] + t_{ijkm} \quad (4.1)$$

$$MS = \max [C_{ijkm}] \quad (4.2)$$

Tahap *Local Search*: terdiri dari proses *exchange* dan *insert*.

#### Langkah 8

Nyatakan jadwal yang terbentuk sebagai *S* dan *makespan* yang dihasilkan sebagai *MS*.

#### Langkah 9

Hitunglah nilai  $MS_{Th}$ .

$$MS_{Th} = MS \text{ terbaik} + (MS \text{ terbaik} + \% Th) \quad (4.3)$$

#### Langkah 10 (Proses Exchange)

Set:  $iter = 1$

#### Langkah 11

Tentukan lintasan kritis, kemudian pilih salah satu lintasan yang memberikan pertukaran pasangan operasi terbanyak. Nyatakan lintasan kritis yang terbentuk sebagai *LK*.

Set: *LK*

#### Langkah 12

Tentukan himpunan pasangan operasi yang dipertukarkan sebagai *W*, dan hitung jumlah pasangan operasi yang dipertukarkan sebagai *X*.

#### Langkah 13

Set:  $Y = 1$

Tentukan dua operasi yang akan dipertukarkan (operasi a dan operasi b) yang salah satu operasinya berada pada lintasan kritis dan kedua operasi tersebut berada pada stasiun kerja yang sama. Selain itu, antara operasi a dan operasi b saling berhimpit satu sama lain.

#### Langkah 14

Tentukan himpunan operasi yang perlu dijadwalkan kembali yaitu  $\Omega PM[a], \Omega PM[b], \Omega PJ[a], \Omega PJ[b]$

#### Langkah 15

Tentukan *ready time* operasi a dan operasi b yang dipertukarkan.

- Operasi a

Nyatakan *ready time* operasi  $a \geq \text{completion time } \Omega PM[b] \text{ dan } \Omega PJ[a]$ .

Jika  $\Omega PJ[a]$  terdapat operasi 3, maka untuk operasi 3 dapat dijadwalkan setelah operasi 2 pada  $\Omega PJ[a]$  atau setelah operasi a terjadwalkan dengan tidak melebihi *completion time* dari *job-job* sebelumnya pada mesin yang sama di stasiun kerja yang sama pada operasi 3 tersebut.

- Operasi b

Nyatakan *ready time* operasi  $b \geq \text{completion time } \Omega PM[a] \text{ dan } \Omega PJ[b]$ .

Jika  $\Omega PJ [b]$  terdapat operasi 3, maka untuk operasi 3 dapat dijadwalkan setelah operasi 2 pada  $\Omega PJ [b]$  atau setelah operasi b terjadwalkan dengan tidak melebihi *completion time* dari *job-job* sebelumnya pada mesin yang sama di stasiun kerja yang sama pada operasi 3 tersebut.

#### Langkah 16

Hitung saat selesai tiap operasi job dan makespan setelah dilakukan exchange.

$$C_{jkm(iter)} = maks [r_{ij}, R_{km}] + t_{jkm} \quad (4.4)$$

Tetapkan  $MS_{iter}$

$$MS_{iter} = maks [C_{jkm(iter)}] \quad (4.5)$$

Tampilkan *gantt chart* proses *exchange*.

#### Langkah 17

Periksa apakah  $MS_{iter} \leq MS_{Th}$

Jika ya, lanjutkan ke langkah 18.

Tetapkan  $MS_{iter}$

Set:  $iter = iter + 1$

Jika tidak, kembali ke langkah 13

Set:  $Y = Y + 1$

#### Langkah 18

Perbaharui nilai  $MS_{Th}$  dengan menggunakan persamaan (4.4).

#### Langkah 19

Periksa apakah  $W \neq \{\}$

Jika ya, kembali ke langkah 13.

Jika tidak, lanjutkan ke langkah 20.

#### Langkah 20 (Proses *Insert*)

Inputkan:  $S'$

Tentukan lintasan kritis, nyatakan operasi terakhir yang berada pada lintasan kritis sebagai  $Ob$ . Masukkan operasi-operasi job terakhir yang berada pada lintasan kritis ke dalam himpunan  $Z$ .

#### Langkah 21

Periksa apakah  $Ob$  dapat dilakukan insert.

Jika ya, lakukan *insert* sebanyak *job* yang berada pada posisi  $Ob$  sampai  $Ob_{.1}$  pada operasi-operasi yang terdapat pada lintasan kritis  $Z$  dan lanjutkan ke langkah 22.

Jika tidak, lanjutkan ke langkah 25.

#### Langkah 22

Hitung saat selesai tiap operasi *job* dan makespan setelah dilakukan *insert*.

$$\text{Gunakan persamaan} \quad (4.4)$$

Tetapkan  $MS_{iter}$

$$\text{Gunakan persamaan} \quad (4.5)$$

Tampilkan *gantt chart* proses *insert*.

#### Langkah 23

Periksa  $MS_{iter} \leq MS_{Th}$

Jika ya, gunakan jadwal tersebut dan lanjutkan ke langkah 24.

Tentukan  $Ob$  dan  $Z$



Set:  $iter = iter + 1$   
 Jika tidak, kembali ke langkah 20.

Langkah 24

Periksa apakah  $Z \neq \{\}$ .  
 Jika ya, kembali ke langkah 20.  
 Jika tidak, lanjutkan ke langkah 25.

Langkah 25

Record jadwal yang memberikan *makespan* minimum dan tampilkan gantt chart hasil penjadwalannya.

Hasil penjadwalan yang diperoleh dari hasil penelitian untuk tahap 1 menggunakan kasus 1 homogen dan heterogen dapat dilihat pada Tabel 8 dan Tabel 9.

**Tabel 8. Hasil Penjadwalan *Job Shop* Mesin Paralel Homogen Kasus 1 Menggunakan Algoritma VND with Fixed Threshold**

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	2	4	0	4
	2	2	1	8	10	18
	3	3	2	4	18	22
B	1	2	2	5	0	5
	2	1	2	6	5	11
	3	3	2	2	11	13
C	1	1	1	2	0	2
	2	2	1	6	4	10
	3	3	1	9	10	19
D	1	2	1	4	0	4
	2	1	1	9	4	13
	3	3	2	2	13	15
E	1	2	2	6	5	11
	2	1	2	7	11	18
	3	3	1	2	19	21

**Tabel 9. Hasil Penjadwalan *Job Shop* Mesin Paralel Heterogen Kasus 1 Menggunakan Algoritma VND with Fixed Threshold**

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	2	4	0	4
	2	2	2	5	5	10
	3	3	2	1	10	11
B	1	2	2	1	0	1
	2	1	1	5	1	6
	3	3	2	2	6	8
C	1	1	1	1	0	1
	2	2	2	4	1	5
	3	3	1	9	5	14
D	1	2	1	3	0	3
	2	1	2	8	6	14
	3	3	1	2	14	16
E	1	2	1	6	3	9
	2	1	2	3	9	12
	3	3	2	4	12	16

Hasil penjadwalan yang diperoleh dari hasil penelitian untuk tahap 2 menggunakan kasus 2 dan kasus 3 homogen dan heterogen dapat dilihat pada Tabel 10 sampai dengan Tabel 13.

**Tabel 10. Hasil Penjadwalan Job Shop Mesin Paralel Homogen Kasus 2 Menggunakan Algoritma VND with Fixed Threshold**

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	2	5	0	5
	2	2	1	2	7	9
	3	3	2	6	9	15
B	1	2	1	2	5	7
	2	1	2	4	11	15
	3	3	3	6	15	21
C	1	1	1	3	0	3
	2	2	2	5	3	8
	3	3	1	6	8	14
D	1	2	2	3	0	3
	2	1	1	8	3	11
	3	3	1	7	14	21
E	1	2	1	5	0	5
	2	1	2	7	5	11
	3	3	2	8	11	19

**Tabel 11. Hasil Penjadwalan Job Shop Mesin Paralel Homogen Kasus 3 Menggunakan Algoritma VND with Fixed Threshold**

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	2	6	0	6
	2	2	1	7	6	13
	3	3	1	6	13	19
B	1	2	3	8	0	8
	2	1	1	5	8	13
	3	3	2	3	13	16
C	1	1	1	3	0	3
	2	2	1	3	3	6
	3	3	1	4	6	10
D	1	2	2	7	0	7
	2	1	2	4	11	15
	3	3	2	3	16	19
E	1	2	1	3	0	3
	2	1	1	8	3	11
	3	3	1	6	11	17

**Tabel 12. Hasil Penjadwalan Job Shop Mesin Paralel Heterogen Kasus 2 Menggunakan Algoritma VND with Fixed Threshold**

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	2	4	0	4
	2	2	2	2	4	6
	3	3	2	4	6	10
B	1	2	1	1	0	1
	2	1	2	1	1	2
	3	3	1	4	2	6
C	1	1	1	3	4	7
	2	2	2	3	7	10
	3	3	2	3	10	13
D	1	2	2	2	0	2
	2	1	1	6	7	13
	3	3	3	4	13	17
E	1	2	1	2	2	4
	2	1	2	4	4	8
	3	3	2	6	8	14

**Tabel 13. Hasil Penjadwalan Job Shop Mesin Paralel Heterogen Kasus 3 Menggunakan Algoritma VND *with Fixed Threshold***

Job	Operasi	SK	Mesin Paralel	Waktu Operasi	Start Time	Finish Time
A	1	1	1	1	0	1
	2	2	1	1	2	3
	3	3	1	2	3	5
B	1	2	2	5	0	5
	2	1	2	3	5	8
	3	3	2	1	8	9
C	1	1	2	3	0	3
	2	2	1	1	3	4
	3	3	1	2	4	6
D	1	2	1	2	0	2
	2	1	1	3	8	11
	3	3	2	1	11	12
E	1	2	3	2	0	2
	2	1	2	6	2	8
	3	3	3	1	8	9

Hasil perbandingan algoritma usulan dengan penelitian yang telah dilakukan sebelumnya dapat dilihat pada Tabel 14 dan Tabel 15.

**Tabel 14. Hasil Perbandingan *Makespan* untuk Mesin Paralel Homogen**

Kasus	Jumlah Job	SK			<i>Makespan</i>			
		1	2	3	Puryani (2003)	Luis et.al (2003)	Putra (2010)	Algoritma Usulan
Kasus 1	5	2	2	2	23	22	22	22
Kasus 2		2	2	3	23	22	21	21
Kasus 3		2	3	3	19	19	19	19

**Tabel 15. Hasil Perbandingan *Makespan* untuk Mesin Paralel Heterogen**

Kasus	Jumlah Job	SK			<i>Makespan</i>			
		1	2	3	Puryani (2003)	Luis et.al (2003)	Putra (2010)	Algoritma Usulan
Kasus 1	5	2	2	2	18	17	16	16
Kasus 2		2	2	3	17	17	17	17
Kasus 3		2	3	3	12	12	12	12

## 4.2 Analisis

Berdasarkan Tabel 14 dan Tabel 15 dapat diketahui bahwa algoritma VND *with fixed threshold* untuk mesin paralel homogen dan heterogen untuk kasus 1 sampai dengan kasus 3 menghasilkan nilai yang sama dengan Puryani (2003), Luis et. al. (2003), dan Putra (2010). Berdasarkan hasil dari tahap 2, maka algoritma usulan memiliki keandalan untuk menyelesaikan permasalahan *job shop* kelompok mesin paralel homogen dan heterogen. Nilai *threshold* yang digunakan sebesar 10%, memberikan nilai *makespan* pada mesin paralel homogen untuk kasus 1 22 satuan waktu, kasus 2 21 satuan waktu, dan kasus 3 19 satuan waktu. Sedangkan pada mesin paralel heterogen untuk kasus 1 16 satuan waktu, kasus 2 17 satuan waktu, dan kasus 3 12 satuan waktu. Nilai *threshold* ini berpengaruh pada besar penerimaan ruang solusi dan memberikan pengaruh juga pada jumlah iterasi yang akan dilakukan.

## 6. KESIMPULAN

### 6.1 Kesimpulan

Beberapa kesimpulan dari hasil penelitian ini diantaranya:

1. Pengembangan algoritma yang diusulkan adalah algoritma *variable neighborhood descent with fixed threshold* yang diujikan untuk menyelesaikan masalah pada kelompok mesin paralel homogen dan heterogen untuk minimasi jumlah *makespan*.
2. Pengujian pada tahap-tahap yang dilakukan menunjukkan bahwa algoritma usulan dapat digunakan untuk menyelesaikan permasalahan pada kelompok mesin paralel homogen dan heterogen.
3. Algoritma usulan memungkinkan ruang penerimaan solusi yang lebih besar dibandingkan penelitian-penelitian sebelumnya, karena mempertimbangkan semua solusi yang muncul meskipun bukan solusi terbaik selama solusi tersebut berada dalam ambang batas penerimaan.
4. Hasil yang diperoleh dari algoritma usulan sama dengan penelitian sebelumnya yang dilakukan oleh Putra (2010).

### 6.2 Saran

Beberapa penelitian selanjutnya yang dapat dilakukan berbasis penelitian ini diantaranya:

1. Penelitian selanjutnya dapat dilakukan pada kelompok mesin paralel dengan jumlah *job* yang lebih besar dan beberapa kasus yang telah diketahui dengan menggunakan program untuk meminimasi kesalahan pada saat melakukan penjadwaalan.
2. Penelitian selanjutnya dapat menggunakan algoritma metaheuristik lain yang lebih kompleks.

## REFERENSI

Dueck, G. Dan Scheuer, T. (1990). *Threshold Accepting. A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing*. Journal Of Computational Physics 90, 161-175.

Luis, M., Zaini, E., Prasetyo, H., Rosidi, D.S. (2004). *Model Penjadwalan Job Shop Dengan Kelompok Mesin Paralel Homogen Menggunakan Algoritma Ant Colony System Untuk Minimasi Makespan*, Prosiding Simposium Nasional II RAPI, I-108-I-115.

Mladenovic, N. Dan Hansen, P,. (1997). *Variable Neighborhood Search*, Computer Operation Research 24, 1097-1100.

Puryani. (2003). *Model Penjadwalan Job Shop Kelompok Mesin Paralel Menggunakan Algoritma Jadwal Aktif dan Algoritma Branch And Bound Like (Babl) Untuk Kriteria Minimasi Makespan*. Tugas Sarjana – Program Studi Teknik Industri, Institut Teknologi Nasional, Bandung.

Putra, H.P. (2010). *Algoritma Penjadwalan Kelompok Mesin Paralel Homogen dan Heterogen Menggunakan Greedy Randomized Adaptive Search Procedure Untuk Kriteria Minimasi Makespan* Tugas Sarjana – Program Studi Teknik Industri, Institut Teknologi Nasional, Bandung.