

Bab IV

Hasil dan Pembahasan

4.1. Deskripsi masalah penjadwalan alternatif mesin

Masalah penjadwalan yang dibahas dalam penelitian ini adalah masalah penjadwalan alternatif mesin. Misalkan diberi suatu himpunan job I . . Tiap job $i \in I$ mempunyai due date D_i dan bobot kepentingan w_i . Misal H menyatakan himpunan alternatif mesin. Setiap jenis mesin h pada slot k mempunyai ketersediaan H_{hk} . $H_i \subseteq H$, $H_i \neq \emptyset$ menyatakan himpunan alternatif mesin yang dapat mengerjakan job i . Tiap job i dapat dikerjakan pada salah satu mesin $h \in H_i$. Waktu operasi job i pada mesin h dinyatakan dengan matriks t_{ih} . Horison waktu dideskritisasi dalam K satuan waktu. Pengerjaan tiap job i membutuhkan operator. Ketersediaan operator pada slot k dinyatakan dengan P_k .

Misalkan ada 6 job 4 mesin dan masing-masing job ditentukan *due date* dan bobotnya. Horison waktu didiskretisasi dalam 8 satuan waktu. Permasalahan tersebut ditabelkan dalam Tabel 4.1 dan Tabel 4.2 serta digambarkan dalam Gambar 4.1.

Tabel 4.1. Permasalahan penjadwalan alternatif mesin

Job	Waktu pengerjaan job untuk tiap mesin (hari)				Due date (hari)	Bobot	Horison waktu (hari)
	1	2	3	4			
1	t_{11}	t_{12}	-	-	D_1	W_1	k
2	t_{21}	t_{22}	-	-	D_2	w_2	
3	-	-	t_{33}	t_{34}	D_3	w_3	
4	-	-	t_{43}	t_{44}	D_4	W_4	
5	t_{51}	t_{52}	-	-	D_5	w_5	
6	-	-	t_{63}	t_{64}	D_6	w_6	

3.2. Pendekatan pemecahan masalah

Pemecahan masalah penjadwalan alternatif mesin mempertimbangkan mesin paralel dan sumber serentak yang dikembangkan menggunakan dua pendekatan, yaitu pendekatan optimal dan pendekatan heuristik. Pendekatan

optimal yang dikembangkan adalah model pemrograman linier bilangan bulat. Sementara itu, pendekatan heuristik yang dikembangkan adalah Algoritma heuristik. Adapun rumusan matematik yang digunakan berbasis pada rumusan matematik horison waktu diskret.

Berdasarkan pendekatan pemecahan masalah, model yang dikembangkan ada dua, yaitu :

- a. Model Pemrograman linier bilangan bulat (*ILP*) untuk pemecahan masalah penjadwalan alternatif mesin mempertimbangkan mesin paralel dan sumber serentak berbasis rumusan matematik horison waktu diskret
- b. Algoritma heuristik untuk pemecahan masalah penjadwalan alternatif mesin mempertimbangkan mesin paralel dan sumber serentak berbasis rumusan matematik horison waktu diskret

3.1. Model *ILP* untuk pemecahan masalah penjadwalan alternatif mesin

3.2.1. Model penjadwalan untuk job majemuk operasi tunggal

3.2.1.1. Notasi

- a. Himpunan

H_i = himpunan mesin alternatif untuk mengerjakan job i .

- b. Indeks

i = job

h = mesin

k = slot waktu

- c. Parameter

N = jumlah job

W_i = bobot job i

D_i = *Due date* job i

K = horison waktu

t_{ih} = waktu pengerjaan job i pada mesin h , dimana $h \in H_i$

H_{hk} = konstanta yang menunjukkan ketersediaan mesin h pada slot k .

P_k = ketersediaan operator pada slot waktu k

- d. Variabel keputusan

C_i = saat selesai pengerjaan job i .

C_{ih} = saat selesai pengerjaan job i pada mesin h

X_{ih} = variabel biner 0-1 yang menyatakan bahwa job i dikerjakan oleh mesin h

C_{ihk} = variabel biner 0-1, bernilai 1 jika job i yang dikerjakan pada mesin h diselesaikan pada slot waktu k , bernilai 0 bila tidak.

T_i = *Tardiness* job i

E_i = *Earliness* job i

t_i = waktu pengerjaan job i

X_{ihk} = variabel biner 0-1, bernilai 1 jika job i pada slot waktu k menggunakan mesin h , bernilai 0 bila untuk yang lain

B_i = Saat mulai pengerjaan job i .

e. Ukuran performansi

Z = *total tardiness* tertimbang

3.2.1.2. Asumsi

Asumsi model adalah sebagai berikut :

1. Semua job dianggap tersedia pada waktu $k=1$
2. Waktu pengerjaan job sudah mencakup waktu *set up*.
3. Pengerjaan suatu job *nonpreemptive*.
4. Horison waktu K dianggap cukup panjang untuk menyelesaikan semua job, $C_i \leq K$ untuk semua i .
5. Setiap job dikerjakan secara simultan antara satu mesin dan satu operator

3.2.1.3. Perumusan model

Perumusan fungsi tujuan :

Fungsi tujuan adalah meminimasi *total tardiness* tertimbang

$$\text{Minimasi } Z = \sum_i w_i T_i \quad (3.1)$$

Perumusan fungsi pembatas:

- a. Pembatas fungsi *tardiness*.

$$T_i - E_i - C_i + D_i = 0, \forall i \quad (3.2)$$

- b. Pembatas yang menyatakan bahwa saat selesai pengerjaan job adalah lebih besar atau sama dengan waktu pengerjaannya.

$$C_i \geq t_i, \forall i \quad (3.3)$$

- c. Pembatas waktu pengerjaan job

$$t_i = \sum_h t_{ih} X_{ih}, \forall i; \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.4)$$

- d. Pembatas penjamin bahwa setiap job hanya dikerjakan oleh satu mesin

$$\sum_h X_{ih} = 1, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.5)$$

- e. Pembatas saat selesai pengerjaan job pada mesin h . Pembatas ini menyatakan bahwa saat selesai pengerjaan job i pada mesin h adalah lebih besar atau sama dengan waktu pengerjaan job i pada mesin h

$$C_{ih} \geq t_{ih}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.6)$$

$$\sum_h C_{ih} = C_i, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.7)$$

- f. Pembatas yang menentukan waktu akhir penyelesaian setiap job pada masing-masing mesin. Karena c_{ihk} sama dengan nol kecuali dalam saat waktu akhir penyelesaian, maka penjumlahan pada ruas kanan adalah sama dengan k kali 1, dimana k adalah waktu akhir penyelesaian

$$\sum_k C_{ihk} k = C_{ih}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.8)$$

- g. Pembatas penjamin bahwa job i pada tiap mesin h hanya diselesaikan sekali sepanjang horison waktu.

$$\sum_k C_{ihk} = 1, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.9)$$

- h. Pembatas bilangan biner untuk menjamin secara mutlak hanya satu mesin yang terpilih dalam pengerjaan job i

$$X_{ih} \in \{0,1\}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.10)$$

- i. Pembatas biner bagi variabel keputusan C_{ihk}

$$C_{ihk} \in \{0,1\}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.11)$$

- j. Pembatas penggunaan mesin. Pembatas ini untuk menjamin bahwa jika job tersebut diselesaikan pada slot waktu k , maka antara slot waktu $(k-t_i+1)$ sampai k pengerjaan job tersebut menggunakan mesin h .

$$\sum_{k \leq l \leq k+t_{im}-1} C_{ihl} = X_{ihk}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.12)$$

- k. Pembatas ketersediaan mesin pada tiap slot waktu. Pembatas ini menjamin bahwa pemakaian mesin tidak boleh melebihi jumlah mesin yang tersedia.

$$\sum_i X_{ihk} \leq H_{hk}, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.13)$$

- l. Pembatas biner bagi variabel keputusan X_{ihk}

$$X_{ihk} \in \{0,1\}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.14)$$

- m. Pembatas ketersediaan operator pada setiap slot waktu

$$\sum_i \sum_h X_{ihk} \leq P_k, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (3.15)$$

- n. Pembatas saat awal pengerjaan job.

$$B_i = C_i - t_i + 1, \forall i \quad (3.16)$$

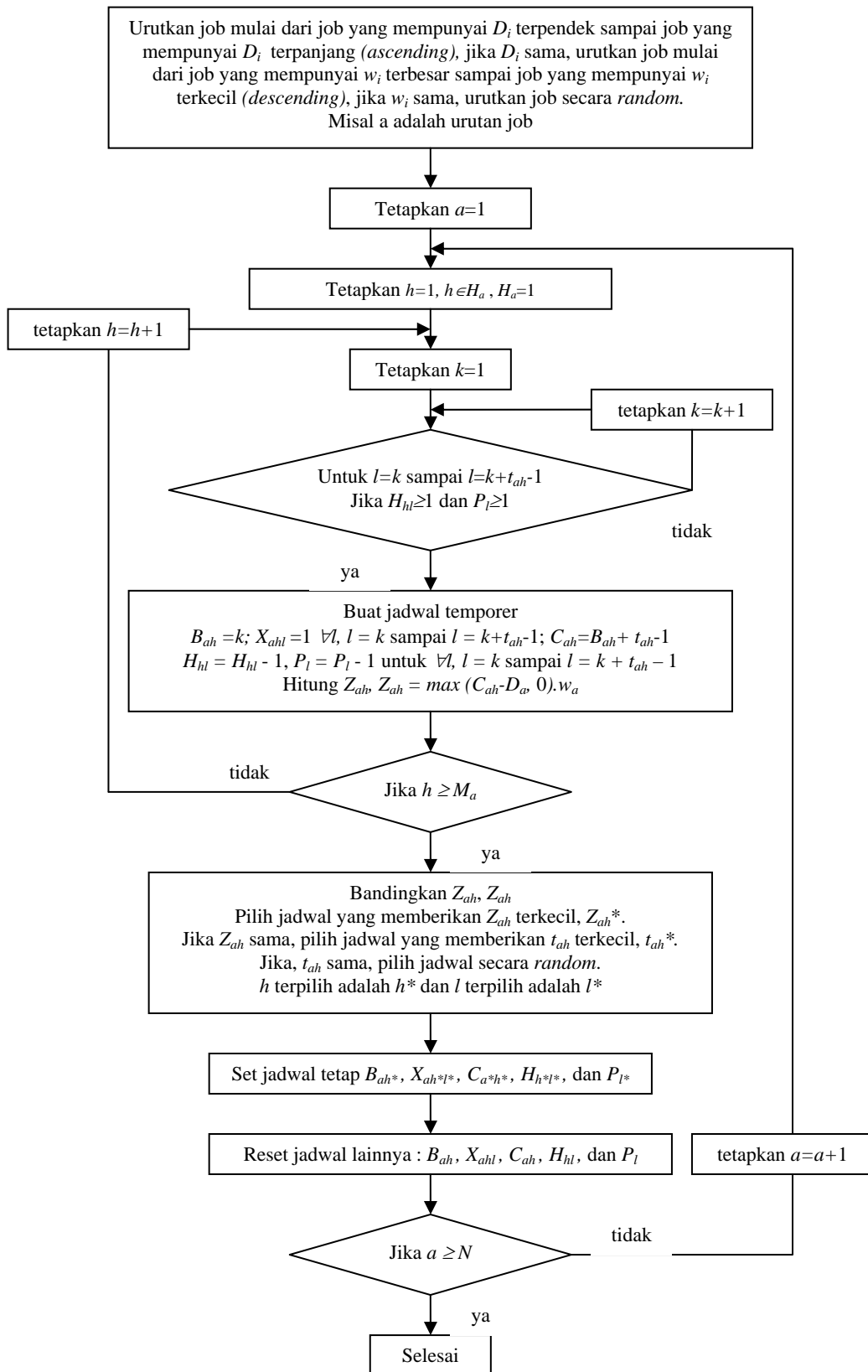
3.2. Algoritma heuristik untuk masalah penjadwalan alternatif mesin

3.3.1. Algoritma penjadwalan untuk job majemuk operasi tunggal

Algoritma penjadwalan untuk job majemuk operasi tunggal adalah sebagai berikut:

1. Urutkan job mulai dari job yang mempunyai D_i terpendek sampai job yang mempunyai D_i terpanjang (ascending).
Jika D_i sama, urutkan job mulai dari job yang mempunyai w_i terbesar sampai job yang mempunyai w_i terkecil (descending).
Jika w_i sama, urutkan job secara *random*.
Misal a menyatakan urutan job.
2. Tetapkan $a=1$
3. Tetapkan $h=1, h \in H_a, H_a = 1$
4. Tetapkan $k=1$
5. Untuk $l=k$ sampai $l=k+ t_{ah}-1$ jika $H_{hl} \geq 1$ dan $P_l \geq 1$, lanjutkan ke langkah 6.
Jika tidak, tetapkan $k=k+1$ lalu ulangi langkah 5
6. Buat jadwal temporer, tetapkan $B_{ah}=k; X_{ahl} = 1 \quad \forall l, l=k$ sampai $l=k+t_{ah}-1$;
 $C_{ah}=B_{ah}+ t_{ah}-1; H_{hl}=H_{hl}-1, P_l = P_l - 1$ untuk $\forall l, l=k$ sampai $l=k+ t_{ah}-1$;
Hitung $Z_{ah}, Z_{ah}=\max(C_{ah}-D_a, 0) \cdot w_a$ lalu lanjutkan ke langkah 7
7. Jika $h < M_a$, tetapkan $h=h+1, (h \in H_a, H_a=1)$ lalu kembali ke langkah 4. Jika tidak lanjutkan ke langkah 8
8. Bandingkan Z_{ah}, t_{ah} . Pilih jadwal yang memberikan Z_{ah} terkecil, Z_{ah}^* . Jika Z_{ah} sama, pilih jadwal yang memberikan t_{ah} terkecil, t_{ah}^* . Jika, t_{ah} sama, pilih jadwal secara *random*. h terpilih adalah h^* dan l terpilih adalah l^* .
Lanjutkan ke langkah 9
9. Set jadwal tetap : $B_{ah^*}, X_{ah^*l^*}, C_{ah^*}, H_{h^*l^*},$ dan P_{l^*} .
Lanjutkan ke langkah 10
10. Reset jadwal temporer yang lainnya : $B_{ah}, X_{ahl}, C_{ah}, H_{hl},$ dan P_l
Lanjutkan ke langkah 11
11. Jika $a < N$, tetapkan $a=a+1$ lalu kembali ke langkah 3.
Jika tidak hentikan algoritma

Algoritma ini digambarkan dalam *flowchart* pada Gambar 3.3.



Gambar 3.3. Flowchart Algoritma penjadwalan job majemuk operasi tunggal

4.1. Contoh Numerik

Model yang telah dihasilkan diaplikasikan pada 6 contoh numerik, terdiri dari 3 contoh numerik untuk mengaplikasikan masalah job majemuk operasi tunggal dan 3 contoh numerik untuk mengaplikasikan masalah job majemuk operasi majemuk. Hal ini dapat dilihat pada Tabel 4.1.

Tabel 4.1. Penerapan model

Karakteristik model	Penerapan
Job majemuk operasi tunggal	Contoh numerik 1
	Contoh numerik 2
	Contoh numerik 3
Job majemuk operasi majemuk	Contoh numerik 4
	Contoh numerik 5
	Contoh numerik 6

Pada Bab IV ini ditampilkan analisis terhadap penerapan model pada satu contoh numerik untuk masalah job majemuk operasi tunggal (Contoh numerik 3) dan satu contoh numerik untuk masalah job majemuk operasi majemuk (Contoh numerik 4). Tabel 4.2. dan Tabel 4.4. adalah contoh numerik 3 dan 4, sementara Tabel 4.3. dan Tabel 4.5 adalah gambar struktur job contoh numerik 3 dan 4.

Tabel 4.2. Contoh Numerik 3

Job	Waktu pengerjaan job tiap mesin (hari)						Due date (hari)	Bobot	Horison waktu (hari)
	1	2	3	4	5	6			
1	4	3	-	-	-	-	3	1	12
2	4	5	-	-	-	-	4	5	
3	5	3	-	-	-	-	3	2	
4	-	-	3	4	-	-	3	3	
5	-	-	4	5	-	-	4	3	
6	-	-	5	3	-	-	3	4	
7	-	-	4	5	-	-	4	4	
8	-	-	-	-	5	3	3	2	
9	-	-	-	-	3	4	3	5	
10	-	-	-	-	3	5	3	1	
Mesin paralel	1	2	2	2	2	1			
Jumlah operator	8								

Tabel 4.3. Struktur job contoh numerik 3

Job	Urutan operasi	Job	Urutan operasi
1	①	2	①

3	①	4	①
5	①	6	①
7	①	8	①
9	①	10	①

4.2. Penerapan Model *ILP* untuk masalah penjadwalan alternatif mesin

4.2.1. Penerapan Model *ILP* untuk penjadwalan job majemuk operasi tunggal

Contoh numerik yang dipakai untuk penerapan model ini adalah contoh numerik 3. Solusi penjadwalan dalam matriks 0-1 ditampilkan pada Tabel 4.6 dan solusi penjadwalan ditampilkan pada Tabel 4.7.

Tabel 4.6. Solusi penjadwalan contoh numerik 3 dalam matrik 0-1 (Model *ILP*)

Mesin	Job	Horison waktu (hari)											
		1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	0	0	0	0	0	0	0	0	0	0
	2	1	1	1	1	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0	0
Mesin terpakai		1	1	1	1	0	0	0	0	0	0	0	0
Mesin tersedia		1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0

Tabel 4.6. Solusi penjadwalan contoh numerik 3 dalam matrik 0-1 (lanjutan)

	8	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0	0
Mesin terpakai		1	1	1	1	1	1	0	0	0	0	0	0

Mesin tersedia	1	1	1	1	1	1	1	1	1	1	1	1
3	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	1	1	1	1	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0
	7	1	1	1	1	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0
Mesin terpakai	2	2	2	2	0	0	0	0	0	0	0	0
Mesin tersedia	1	1	1	1	1	1	1	1	1	1	1	1
4	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0
	4	1	1	1	1	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0
Mesin terpakai	2	2	2	1	0	0	0	0	0	0	0	0
Mesin tersedia	1	1	1	1	1	1	1	1	1	1	1	1
5	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	0	0	0	0	0
	10	0	0	0	1	1	1	0	0	0	0	0
Mesin terpakai	1	1	1	1	1	1	0	0	0	0	0	0
Mesin tersedia	1	1	1	1	1	1	1	1	1	1	1	1
6	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0
	8	1	1	1	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0

Tabel 4.6. Solusi penjadwalan contoh numerik 3 dalam matrik 0-1 (lanjutan)

Mesin terpakai	1	1	1	0	0	0	0	0	0	0	0	0
Mesin tersedia	1	1	1	1	1	1	1	1	1	1	1	1
Operator terpakai	8	8	8	6	2	2	0	0	0	0	0	0
Operator tersedia	8	8	8	8	8	8	8	8	8	8	8	8

Tabel 4.7. Solusi penjadwalan contoh numerik 3 dengan Model *ILP*

Mesin	Mesin paralel	Horison waktu											
		1	2	3	4	5	6	7	8	9	10	11	12
1	1	2											
2	1	3		1									
	2												
3	1	5											
	2	7											
4	1	4											
	2	6											
5	1	9			10								
	2												
6	1	8											
Jumlah operator		8	8	8	6	2	2	0	0	0	0	0	0
Jumlah operator tersedia		8	8	8	8	8	8	8	8	8	8	8	8

Tabel 4.6 dan 4.7 di atas menunjukkan bahwa solusi penjadwalan *feasible*, dimana tidak ada job dan mesin yang tumpang tindih serta jumlah mesin dan operator terpakai pada slot waktu $k \leq$ jumlah mesin dan operator yang tersedia.

4.3. Penerapan Algoritma heuristik

4.3.1. Penerapan Algoritma heuristik untuk penjadwalan job majemuk operasi tunggal

Contoh numerik yang dipakai untuk penerapan model ini adalah contoh numerik 3. Solusi penjadwalan ditampilkan pada Tabel 4.10.

Tabel 4.10. Solusi penjadwalan contoh numerik 3 dengan Algoritma heuristik

Jenis mesin	Mesin paralel	Horison waktu											
		1	2	3	4	5	6	7	8	9	10	11	12
1	1				2	2	2	2					
2	1	1	1	1									
	2	3	3	3									
3	1	4	4	4	7	7	7	7					
	2	5	5	5	5								
4	1	6	6	6									
	2												
5	1	10	10	10									
	2	9	9	9									
6	1	8	8	8									
Jumlah operator bekerja		8	8	8	3	2	2	2					

Jumlah operator tersedia	8	8	8	8	8	8	8	8	8	8	8	8
--------------------------	---	---	---	---	---	---	---	---	---	---	---	---

4.4. Rekapitulasi hasil

Penerapan model *ILP* dilakukan menggunakan komputer intel pentium III 550 MHz dengan RAM 192 MHz dan rekapitulasi hasil penjadwalan dengan model *ILP* ini ditampilkan pada Tabel 4.11. Adapun penerapan Algoritma heuristik dilakukan secara manual dan rekapitulasi hasil penjadwalan dengan Algoritma heuristik ini ditampilkan pada Tabel 4.12.

Perbandingan nilai fungsi tujuan antara Model *ILP* dan Algoritma Heuristik ditampilkan pada Tabel 4.13.

Tabel 4.12. Rekapitulasi hasil penjadwalan dengan Model *ILP*

Contoh Numerik	Karakteristik Model	Nilai Fungsi tujuan	waktu komputasi (detik)
1	Job majemuk operasi tunggal	0	20
2	Job majemuk operasi tunggal	0	27
3	Job majemuk operasi tunggal	9	21

Tabel 4.13 Rekapitulasi hasil penjadwalan dengan Algoritma Heuristik

Contoh numerik	Karakteristik model	Nilai fungsi tujuan
1	Job majemuk operasi tunggal	0
2	Job majemuk operasi tunggal	1
3	Job majemuk operasi tunggal	28

Tabel 4.14. Perbandingan nilai fungsi tujuan antara Model *ILP* dan Algoritma Heuristik

Contoh numerik	Nilai fungsi tujuan		Perbandingan nilai fungsi tujuan
	Model ILP	Algoritma Heuristik	

1	0	0	0
2	0	1	0
3	9	28	2,111

Berdasarkan perbandingan nilai fungsi tujuan antara Model *ILP* dan Algoritma Heuristik tersebut, tampak bahwa Algoritma Heuristik yang dikembangkan bisa memberikan nilai yang sama dengan Model *ILP* pada contoh numerik 1. Pada contoh numerik yang lain, Model *ILP* yang dikembangkan memberikan nilai fungsi tujuan yang lebih baik dari pada Algoritma heuristik.