BAB II
TINJAUAN PUSTAKA

Review Penelitian Sebelumnya

2.1. Pengembangan model matematik horison waktu diskret optimal untuk penjadwalan job banyak operasi tunggal pada mesin alternatif [Sukendar, 2007]

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 = *Duedate* 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

Asumsi

- Semua job dianggap tersedia pada waktu $k=1$
- Waktu pengerjaan job sudah mencakup waktu *set up*.
- Pengerjaan suatu job *nonpreemptive.*
- Horison waktu *K* dianggap cukup panjang untuk menyelesaikan semua job, $C_i \leq K$ untuk semua *i*.

Perumusan model

Perumusan fungsi tujuan :

Fungsi tujuan adalah minimisasi *total tardiness* tertimbang

$$Minimasi\ \mathrm{Z} = \sum_i w_i T_i \qquad (1)$$

Perumusan fungsi pembatas:

a. Pembatas fungsi *tardiness*.

$$T_i - E_i - C_i + D_i = 0\,, \forall i \qquad (2)$$

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

$$C_i \geq t_i\,, \forall i \qquad (3)$$

c. Pembatas waktu pengerjaan job

$$t_i = \sum_h t_{ih} X_{ih}, \forall i; \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (4)$$

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

$$\sum_h X_{ih} = 1, \forall i, \forall h, h \in H_i; H_i \subseteq H, H_i \neq \phi \quad (5)$$

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*

$$C_{ih} \geq t_{ih}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (6)$$

$$\sum_h C_{ih} = C_i, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (7)$$

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

$$\sum_k C_{ihk} k = C_{ih}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (8)$$

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

$$\sum_k C_{ihk} = 1, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (9)$$

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

$$X_{ih} \in \{0,1\}, \forall i, \forall h; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (10)$$

i. Pembatas biner bagi variabel keputusan C_{ihk}

$$C_{ihk} \in \{0,1\}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (11)$$

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*.

$$\sum_{k \leq l \leq k + t_{im} - 1} C_{ihl} = X_{ihk}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \quad (12)$$

k. Pembatas ketersediaan mesin pada tiap slot waktu. Pembatas ini menjamin bahwa pemakaian mesin tidak boleh melebihi satu.

$$\sum_i X_{ihk} \le H_{hk}\ , \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \qquad (13)$$

l. Pembatas biner bagi variabel keputusan X_{ihk}

$$X_{ihk} \in \{0,1\}, \forall i, \forall h, \forall k; h \in H_i; H_i \subseteq H; H_i \neq \phi \qquad (14)$$

m. Pembatas saat awal pengerjaan job.

$$B_i = C_i - t_i + 1, \quad \forall i \qquad (15)$$

2.1. Pengembangan model matematik horison waktu diskret heuristik untuk penjadwalan job banyak operasi tunggal pada mesin alternatif [Sukendar, 2007]

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 = *Duedate* 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

Asumsi

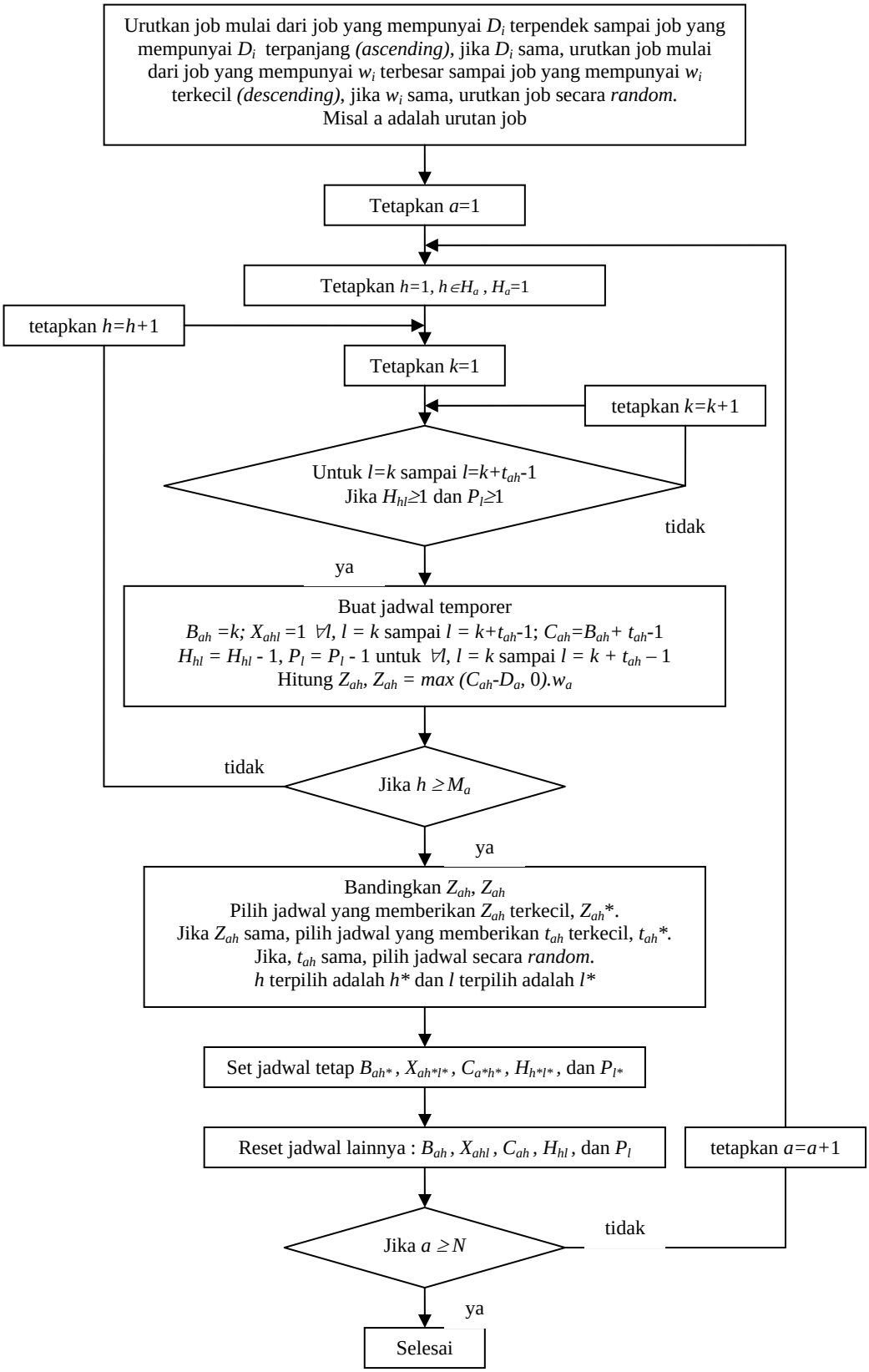
- Semua job dianggap tersedia pada waktu k=1
- Waktu pengerjaan job sudah mencakup waktu *set up*.
- Pengerjaan suatu job *nonpreemptive.*
- Horison waktu *K* dianggap cukup panjang untuk menyelesaikan semua job, $C_i \leq K$ untuk semua *i*.

<u>Algoritma penjadwalan</u> :

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 $B_{ah}=k$; $X_{ahl}=1 \ \forall l$, $l=k$ 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}, $Z_{ah}=max(C_{ah}-D_a, 0).w_a$ lalu lanjutkan ke lngkah 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 2.



Gambar 2. *Flowchart* Algoritma penjadwalan job majemuk operasi tunggal