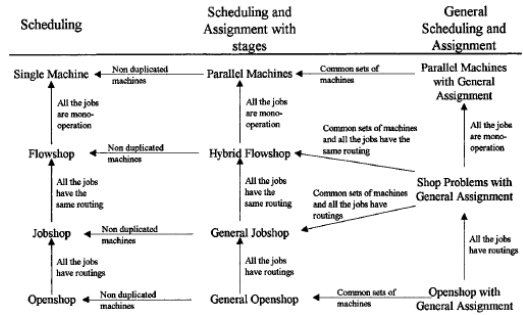


# Job Shop Scheduling Problems

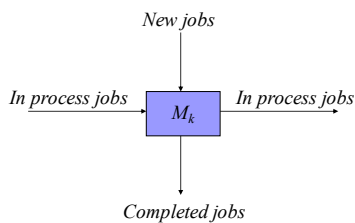
Mata Kuliah: Penjadwalan Produksi  
Teknik Industri – Universitas Brawijaya

## Scheduling Problems



### Job Shop Scheduling (1)

- Flow shop: aliran kerja unidirectional
- Job shop: aliran kerja tidak unidirectional



3

### Job Shop Scheduling (2)

- Flow shop: Indeks  $(i, j)$

Waktu proses

Job	$M_1$	$M_2$	...	$M_m$
$J_1$	...	...	...	...
$J_2$	...	...	...	...
$\vdots$	...	...	...	...
$J_n$	...	...	...	...

- Job shop: Indeks  $(i, j, k)$
- $\swarrow$        $\downarrow$        $\searrow$   
 job    operasi    mesin

$t_{ijk}$  Waktu operasi ke  $j$  untuk pemrosesan job  $i$  di mesin  $k$

4

### Job Shop Scheduling (3)

Waktu Proses				Routing			
	Operasi 1	Operasi 2	Operasi 3		Operasi 1	Operasi 2	Operasi 3
Job J1	4	3	2	Job J1	1	2	3
Job J2	1	4	4	Job J2	2	1	3
Job J3	3	2	3	Job J3	3	2	1
Job J4	3	3	1	Job J4	2	3	1

$t_{233} = 4$        $t_{313} = 3$        $1 = t_{431}$

5

### Geser-Kiri (left-shift)

- Geser-kiri lokal (*local left-shift*)  
penyesuaian (menjadi lebih cepat) saat mulai (*start time*) suatu operasi dengan tanpa mengubah urutan
- Geser-kiri global (*global left-shift*)  
penyesuaian sehingga suatu operasi dimulai lebih cepat tanpa menyebabkan *delay* operasi lain

6

### Jenis Jadwal Pada Job Shop (1)

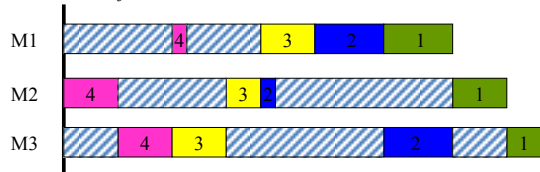
1. Jadwal semiaktif
  - adalah satu set jadwal yang tidak memungkinkan lagi untuk melakukan geser-kiri lokal
  - adalah satu set jadwal yang tidak memiliki *superfluous idle time*  
*Superfluous idle time* terjadi pada jadwal yang apabila suatu operasi dimulai lebih awal tidak menyebabkan perubahan urutan pada mesin manapun

7

### Jenis Jadwal Pada Job Shop (2)

Waktu Proses				Routing			
	Operasi 1	Operasi 2	Operasi 3		Operasi 1	Operasi 2	Operasi 3
Job J1	4	3	2	Job J1	1	2	3
Job J2	1	4	4	Job J2	2	1	3
Job J3	3	2	3	Job J3	3	2	1
Job J4	3	3	1	Job J4	2	3	1

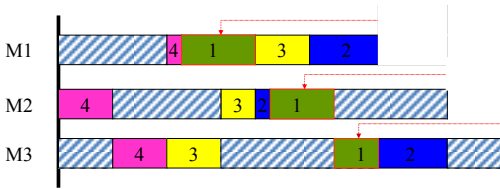
Misal urutan *job* adalah 4-3-2-1:



8

### Jenis Jadwal Pada Job Shop (3)

- Geser-kiri lokal tidak bisa dilakukan (menggeser saat mulai tanpa mengubah urutan)
- Saat mulai operasi (1,1,1) bisa dilakukan tanpa menyebabkan *delay* pada operasi lain (tapi harus mengubah urutan)



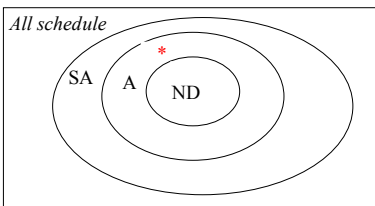
9

### Jenis Jadwal Pada Job Shop (4)

2. Jadwal Aktif  
adalah satu set jadwal yang tidak memungkinkan lagi untuk melakukan geser-kiri global
3. Jadwal *non-delay*  
adalah jadwal aktif yang tidak membiarkan mesin menjadi *idle* bila suatu operasi dapat dimulai

10

### Jenis Jadwal Pada Job Shop (5)



SA = *semiactive*  
A = *active*  
ND = *non-delay*  
\* = *optimal*

11

### Algoritma Pembangkitan Jadwal Aktif (1)

- $PS_t$  = Jadwal parsial yang terdiri  $t$  buah operasi terjadwal
- $S_t$  = Set operasi yang dapat dijadwalkan pada *stage*  $t$ , setelah diperoleh  $PS_t$
- $\sigma_t$  = Waktu tercepat operasi  $j \in S_t$  dapat dimulai
- $\phi_t$  = Waktu tercepat operasi  $j \in S_t$  dapat diselesaikan

12

### Algoritma Pembangkitan Jadwal Aktif (2)

- Step 1. Tentukan  $t=0$ , dan kemudian mulai dengan  $PS_0$  sebagai jadwal parsial nol. Tentukan seluruh operasi tanpa predecessor sebagai  $S_0$ .
- Step 2. Tentukan  $\phi^* = \min_{j \in S_t} \{\phi_j\}$  dan mesin  $m^*$  yaitu mesin tempat  $\phi^*$  dapat direalisasikan
- Step 3. Untuk setiap operasi  $j \in S_t$  yang membutuhkan mesin  $m^*$  dan berlaku  $\sigma_j < \phi^*$ , buat jadwal parsial baru dengan menambahkan operasi  $j$  pada  $PS_t$  dengan saat mulai operasi pada  $\sigma_j$

13

### Algoritma Pembangkitan Jadwal Aktif (3)

- Step 4. Untuk setiap jadwal parsial baru  $PS_{t+1}$ , yang dihasilkan pada Step 3, perbaharui (*up date*) set data berikut:
  - Keluarkan operasi  $j$  dari  $S_t$
  - Tambahkan suksesor langsung operasi  $j$  ke dalam  $S_{t+1}$
  - Naikkan nilai  $t$  dengan 1
- Step 5. Untuk setiap  $PS_{t+1}$  yang dihasilkan pada Step 3, kembali ke Step 2. Lanjutkan langkah-langkah ini sampai suatu jadwal aktif dihasilkan.

14

### Contoh

Waktu Proses

	Operasi 1	Operasi 2	Operasi 3
Job J1	4	3	2
Job J2	1	4	4
Job J3	3	2	3
Job J4	3	3	1

Routing

	Operasi 1	Operasi 2	Operasi 3
Job J1	1	2	3
Job J2	2	1	3
Job J3	3	2	1
Job J4	2	3	1

15

### Jadwal Aktif(1)

Stage	Mesin			$s_t$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_t$
	1	2	3							
0	0	0	0	111	0	4	4			212
				212	0	1	1	1	2	
				313	0	3	3			
				412	0	3	3			
1	0	1	0	111	0	4	4			313
				221	1	4	5			
				313	0	3	3	3	3	
				412	1	3	4			

16

Jadwal Aktif(2)

Stage	Mesin			$s_i$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_i$
	1	2	3							
2	0	1	3	111	0	4	4	4	1	111
				221	1	4	5			
				322	3	2	5			
				412	1	3	4			
3	4	1	3	122	4	3	7			412
				221	4	4	8			
				322	3	2	5			
				412	1	3	4	4	2	

17

Jadwal Aktif(3)

Stage	Mesin			$s_i$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_i$
	1	2	3							
4	4	4	3	122	4	3	7			322
				221	4	4	8			
				322	4	2	6	6	2	
				423	4	3	7			
5	4	6	3	122	6	3	9			423
				221	4	4	8			
				331	4	3	7			
				423	4	3	7	7	3	

18

Jadwal Aktif(4)

Stage	Mesin			$s_i$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_i$
	1	2	3							
6	4	6	7	122	6	3	9			221
				221	4	4	8	8	1	
				331	6	3	9			
				431	7	1	8			
7	8	6	7	122	6	3	9			431
				233	8	4	12			
				331	8	3	11			
				431	8	1	9	9	1	

19

Jadwal Aktif(5)

Stage	Mesin			$s_i$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_i$
	1	2	3							
8	9	6	7	122	6	3	9			122
				233	8	4	12	8	1	
				331	9	3	12			
9	9	9	7	133	9	2	11	11	3	133
				233	8	4	12			
				331	9	3	12			

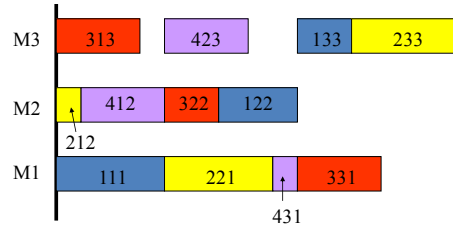
20

Jadwal Aktif(6)

Stage	Mesin			$s_t$	$\sigma_j$	$t_{ij}$	$\phi_j$	$\phi^*$	$m^*$	$PS_t$
	1	2	3							
10	9	9	11	233	11	4	15			331
				331	9	3	12	12	1	
11	12	9	11	233	11	4	15	15	3	233
	12	9	15							

21

Jadwal Aktif(7)



22

### Algoritma Pembangkitan Jadwal *Non-delay*(1)

- Step 1.** Tentukan  $t=0$ , dan kemudian mulai dengan  $PS_0$  sebagai jadwal parsial nol. Tentukan seluruh operasi tanpa *predecessor* sebagai  $S_0$ .
- Step 2.** Tentukan  $\sigma^* = \min_{j \in S_t} \{\sigma_j\}$  dan mesin  $m^*$  yaitu mesin tempat  $\sigma^*$  dapat direalisasikan
- Step 3.** Untuk setiap operasi  $j \in S_t$  yang membutuhkan mesin  $m^*$  dan berlaku  $\sigma_j = \sigma^*$ , buat jadwal parsial baru dengan menambahkan operasi  $j$  pada  $PS_t$  dengan saat mulai operasi pada  $\sigma_j$

23

### Algoritma Pembangkitan Jadwal *Non-delay* (2)

- Step 4.** Untuk setiap jadwal parsial baru  $PS_{t+1}$ , yang dihasilkan pada **Step 3**, perbaharui (*up date*) set data berikut:
- Keluarkan operasi  $j$  dari  $S_t$
  - Tambahkan suksesor langsung operasi  $j$  ke dalam  $S_{t+1}$
  - Naikkan nilai  $t$  dengan 1
- Step 5.** Untuk setiap  $PS_{t+1}$  yang dihasilkan pada **Step 3**, kembali ke **Step 2**. Lanjutkan langkah-langkah ini sampai seluruh jadwal aktif dihasilkan.

24

## Priority Control

- Priority rules: for establishing the priority of orders
- *Common priority rules*
  - First Come First Served (FCFS)
  - Shortest Processing Time (SPT)
  - Earliest Due Date (EDD)
  - Slack per Remaining Operation (S/OPN)
  - Critical Ratio (CR)
  - Queue Ratio (QR)
  - Operation Due Date (OPNDD)

## Referensi

- Introduction to Sequencing and Scheduling. Kenneth R. Baker. Duke University. John Wiley & Sons. 1974.

25

**SELAMAT BELAJAR**