

# PENJADWALAN OPTIMAL TIPE PRODUKSI FLOWSHOP DUA TAHAP MENGGUNAKAN METODE BRANCH AND BOUND DENGAN MEMPERHATIKAN WAKTU TRANSPORTASI

Marie Muhammad<sup>1, a)</sup>, Elis Ratna Wulan<sup>2</sup>

<sup>1,2</sup>Jurusan Matematika, Universitas Islam Negeri Sunan Gunung Djati  
Jln. A.H. Nasution No.150. Kota Bandung

<sup>a)</sup>email: muhammadmarie24@gmail.com

## Abstrak

Penjadwalan produksi dapat dibagi menjadi dua jenis, yaitu penjadwalan produksi tipe *jobshop* dan penjadwalan produksi tipe *flowshop*. Penjadwalan produksi tipe *flowshop* adalah sebuah penjadwalan sebuah produk yang sedemikian rupa sehingga setiap produk diproduksi melalui mesin yang sama dengan alur produksi yang sama. Terdapat beberapa masalah *flowshop*, salah satunya adalah dengan memperhatikan waktu transportasi. Dan metode *Branch and Bound* adalah solusi yang tepat untuk memecahkan masalah penjadwalan produksi dengan memperhatikan waktu transportasi untuk meminimalisir waktu yang terlewat. Pada penulisan Studi Literatur ini, Penjadwalan optimal dari 4 buah *job* dan 2 buah mesin dengan memperhatikan waktu transportasi adalah 1, 2, 4, dan 3 dengan waktu yang terlewat adalah 59 unit satuan waktu.

Kata Kunci : *Flowshop*, *Jobshop*, *Branch and Bound*, Waktu Transportasi, Waktu yang Terlewat

## Pendahuluan

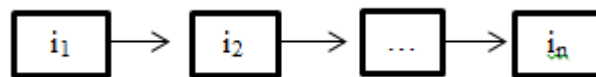
Pemrograman linier merupakan suatu metode yang digunakan untuk melakukan suatu keputusan dalam memecahkan masalah yang bersifat numerik. Penjadwalan merupakan salah satu aspek yang populer di dalam pemrograman linier, tentunya dengan memperhatikan waktu, biaya dan mesin yang akan digunakan. Penjadwalan sangat penting untuk menghasilkan hasil yang optimal sehingga mendapatkan keuntungan yang besar, tanpa terkecuali dalam suatu perusahaan. Permasalahan dalam penjadwalan salah satunya adalah memperhatikan waktu transportasi. Oleh karena itu perlu adanya metode yang cocok untuk masalah penjadwalan dengan memperhatikan waktu transportasi, salah satunya yaitu dengan metode *Branch and Bound*.

## Penjadwalan *Flowshop*

Penjadwalan produksi dapat dibagi menjadi dua jenis, yaitu penjadwalan produksi tipe *jobshop* dan penjadwalan produksi tipe *flowshop*. Penjadwalan produksi *jobshop* adalah penjadwalan sebuah produk yang sedemikian rupa sehingga setiap produksinya melalui mesin produksi yang sama namun setiap produk tidak harus menggunakan alur produksi yang sama. Sementara penjadwalan produksi tipe *flowshop* adalah sebuah penjadwalan sebuah produk yang sedemikian rupa sehingga setiap produk diproduksi melalui mesin yang sama dengan alur produksi yang sama. Keduanya dapat digunakan untuk sebuah proses produksi masal produk yang berbeda dengan menggunakan mesin yang sama, hanya saja alurnya yang berbeda. Namun, tujuan utama dari proses produksi adalah meraih keuntungan terbesar atau hasil yang optimal.

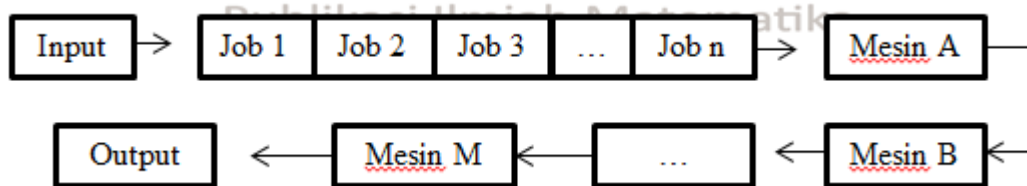
Perbedaan alur menjadi titik pembeda antara dua metode penjadwalan produksi tipe *jobshop* dan penjadwalan produksi tipe *flowshop*. Alur yang dimaksud sendiri adalah sebuah urutan yang mesin yang digunakan dalam sebuah proses produksi.

Secara sistematis, pada umumnya penjadwalan produksi tipe *jobshop* adalah penjadwalan produksi untuk produk yang beragam namun menggunakan mesin yang sama. Sedangkan pada penjadwalan produksi tipe *flowshop*, produk yang diproduksi adalah produk yang beragam namun selain menggunakan mesin yang sama juga memiliki urutan penggunaan mesin yang sama. Namun, kedua penjadwalan produksi tersebut tidak membebaskan untuk setiap produk yang diproduksi harus menggunakan semua mesin yang tersedia. Untuk lebih memahami penjadwalan tipe produksi *flowshop* bisa dilihat pada Gambar 1. dimana penjadwalan setiap job *i* harus melalui mesin yang sama dengan alur yang sama.

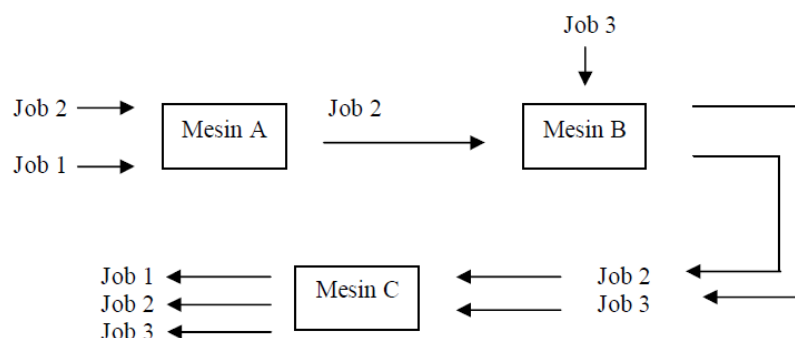


Gambar 1

Penjadwalan produksi tipe *flowshop* sendiri terbagi menjadi dua jenis *flowshop*, yaitu penjadwalan produksi tipe *flowshop* umum dan penjadwalan produksi tipe *flowshop* murni. [7] Perbedaan keduanya dapat dilihat dari mesin produksi yang dilalui oleh setiap produk dalam sebuah proses produksi. Pada penjadwalan produksi tipe *flowshop* murni setiap *job* harus melalui semua mesin produksi yang sama, dengan alur dan urutan yang sama, alur *flowshop* murni dapat dilihat pada Gambar 2. Sedangkan pada penjadwalan produksi tipe *flowshop* umum adalah setiap produk harus melalui mesin produksi yang sama dengan urutan yang sama dan alur yang sama, namun tidak di haruskan melalui semua mesin. Artinya, pada penjadwalan produksi tipe *flowshop* umum jumlah mesin yang digunakan oleh setiap produk tidak diperhatikan asal melalui mesin produksi yang sama dengan alur yang sama. Alur *flowshop* umum dapat dilihat pada Gambar 3. pada Gambar 3 dikhususkan untuk 3 buah *job* dan 3 buah mesin, akan tetapi *flowshop* umum melewati mesin *m* dengan *n* buah *job*, dimana *job* bebas melalui mesin manapun asal mesin produksi dan alurnya sama.



Gambar 2



Gambar 3

Dalam pemilihan penjadwalan produksi yang akan dilakukan, adapun beberapa hal yang harus diperhatikan. Begitu pula ketika kita memilih menggunakan penjadwalan produksi tipe *flowshop*. Ada beberapa syarat yang harus dipenuhi sebelum kita memilih menggunakan penjadwalan produksi tipe

*flowshop*. Asumsi-asumsi tertentu biasanya digunakan dalam sebuah penjadwalan produksi, termasuk pada penjadwalan produksi tipe *flowshop*.

Permasalahan penjadwalan produksi *flowshop* antara lain memiliki asumsi-asumsi berikut ini : [4]

1. Tidak ada operasi pada *job* yang sama berjalan secara bersamaan.
2. Sebuah *job* harus dioperasikan hingga selesai dalam sebuah mesin sebelum dapat dioperasikan pada mesin berikutnya.
3. Setiap *job* harus dikerjakan sampai dengan selesai.
4. Setiap *job* memiliki waktu proses yang konstan atau tetap.
5. Setiap mesin hanya dapat mengoperasikan satu *job* yang sama atau *single tasking*.

### Metode Branch And Bound

Metode *Branch and Bound* merupakan metode pencarian solusi di dalam ruang solusi secara sistematis, yang diimplementasikan ke dalam suatu pohon ruang status dinamis. Pada algoritma ini, masalah digambarkan dalam bentuk diagram pohon dimana masing-masing cabang menggambarkan urutan parsial. Untuk menentukan bagian mana yang menjadi cabang, dihitung *make-span* terendah (*lower bound*) dari masing-masing cabang.

Dalam masalah yang dibahas di sini,  $n$  buah pekerjaan diproses dengan menggunakan  $m$  buah mesin. Setelah sekumpulan pekerjaan diserahkan kepada mesin, pencarian urutan pekerjaan tidak diperlukan lagi dalam mesin tersebut, karena yang menjadi tujuan utama adalah untuk meminimisasi *makespan* dan tidak tergantung dengan waktu pengurutan. Oleh karena itu, algoritma *Branch and Bound* dikembangkan untuk menentukan penyerahan pekerjaan secara optimal kepada mesin. *Branch and Bound* adalah suatu prosedur yang paling umum untuk mencari solusi optimal pada masalah optimasi seperti masalah penjadwalan.

Di dalam algoritma *Branch and Bound*, terdapat tiga buah bagian utama, yaitu : ekspresi batas bawah (*Lower Bound (LB)*), strategi pencarian dan pencabangan (*branching*).

Berikut adalah karakteristik metode *Branch and Bound* untuk meminimumkan masalah : [6]

1. Aturan *Branching* yaitu mencabangkan semua solusi yang mungkin.
2. Aturan *Lower bound (LB)* yaitu menetapkan batas bawah pada setiap solusi yang ada pada aturan *branching*.
3. Strategi pencarian yaitu memilih *node* dari setiap cabang.

Diagram yang menggambarkan proses pencarian solusi disebut pencarian *tree*. Permasalahan yang pertama pada *node* terletak di *tree* paling atas dan disebut *root*. Di dalam prosedur ini, suatu masalah dipecah menjadi beberapa submasalah yang merepresentasikan pembagian kerja secara parsial. Simpul-simpul terus bercabang lebih jauh sampai diperoleh solusi lengkap. Prosedur ini terus diulang sampai pencarian pada pohon berakhir dan solusi optimal ditemukan.

Karena pembahasan dalam Studi Literatur ini dibatasi oleh 2 mesin dan terdiri dari 4 *job*, maka Algoritma yang akan digunakan adalah sebagai berikut: [2]

Langkah 1: hitung ekspetasi waktu proses pada *job*,

$$A_i = a_i \times p_i$$

$$B_i = b_i \times q_i$$

dimana,

$a_i$  : Waktu proses untuk *job*  $i$  di mesin A

$b_i$  : Waktu proses untuk *job*  $i$  di mesin B

$p_i$  : Peluang yang berhubungan dengan waktu proses  $a_i$

$q_i$  : Peluang yang berhubungan dengan waktu proses  $b_i$

$A_i$  : Ekspetasi waktu proses *job i* pada mesin A

$B_i$  : Ekspetasi waktu proses *job i* pada mesin B

Langkah 2: menghitung ekspetasi waktu proses dan waktu transportasi

$$(a) G_i = A_i + t_i \text{ dan } (b) H_i = B_i + t_i$$

dimana,

$A_i$  : Ekspetasi waktu proses *job i* pada mesin A

$B_i$  : Ekspetasi waktu proses *job i* pada mesin B

$t_i$  : Waktu transportasi pada *job i* dari mesin A ke mesin B.

$G_i$  : Jumlah dari ekspetasi waktu proses dan waktu transportasi pada mesin A

$H_i$  : Jumlah dari ekspetasi waktu proses dan waktu transportasi pada mesin B

Langkah 3: Mencari *Lower Bound* (LB), melakukan strategi pencarian dan melakukan pencabangan (*Branching*).

$$l_1 = t(J_r, 1) + \sum_{i \in J_r} G_i + \min_{i \in J_r} H_i$$

$$l_2 = t(J_r, 2) + \sum_{i \in J_r} H_i$$

dimana,

$J_r$  : pemilihan jadwal pada *job r* yang dijadwalkan

$J_r'$  : himpunan tersisa ( $n-r$ ) *job* yang bebas

Langkah 4: membandingkan nilai  $l_1$  dan  $l_2$  untuk kemudian diambil nilai maksimum.  $l = \text{maksimum}(l_1, l_2)$

langkah 5: Setelah mendapatkan nilai maksimum membentuk akar pohon dengan mencari *makespan* terendah dan ulangi langkah 3 dan langkah 4 sampai didapatkan jadwal yang optimal.

Langkah 6: setelah dijadwalkan berdasarkan akar pohon, maka selanjutnya adalah mengurutkan *job* sesuai dengan penjadwalan yang didapatkan dari langkah 5 dan mendapatkan waktu yang terlewati yaitu *makespan* terendah dikurangi total waktu perpindahan antar mesin.

### Analisis

Data yang didapat pada proses produksi ini adalah sebuah produksi dengan 4 buah *job* yang akan dioperasikan pada 2 buah mesin yaitu mesin A dan mesin B. pada mesin A mempunyai waktu proses beserta nilai peluangnya, juga pada mesin B mempunyai waktu proses beserta nilai peluangnya. Bisa dilihat pada tabel 1

Job	Mesin A			Mesin B	
$i$	$a_i$	$p_i$	$t_i$	$b_i$	$q_i$
1	80	0.1	5	40	0.4
2	65	0.2	3	85	0.2
3	20	0.4	6	30	0.1
4	70	0.3	4	30	0.3

Tabel 1

Berdasarkan algoritma dari metode *Branch and Bound*, langkah pertama yaitu mencari nilai ekspetasi dari setiap *job*, lalu kita hitung dengan ketentuan

$$A_i = a_i \times p_i \text{ and } B_i = b_i \times q_i$$

Lalu didapatkan hasil dari perhitungan nilai ekspetasi, bisa dilihat pada tabel 2

Job	Mesin A		Mesin B
$i$	$A_i$	$t_i$	$B_i$
1	8	5	16
2	13	3	17
3	8	6	3
4	21	4	9

Tabel 2

Langkah kedua yaitu mencari nilai  $G_i$  dan  $H_i$ , yang didefinisikan

$$G_i = A_i + t_i$$

$$H_i = B_i + t_i$$

Lalu didapatkan hasil perhitungan dari  $G_i$  dan  $H_i$ , bisa dilihat pada tabel 3

Job	Mesin A	Mesin B
$i$	$G_i$	$H_i$
1	13	21
2	16	20
3	14	9
4	25	13

Tabel 3

Langkah ketiga, yaitu mencari *Lower Bound* (LB), melakukan strategi pencarian dan melakukan *Branching*.

$$l_1 = t(J_r, 1) + \sum_{i \in J_r} G_i + \min_{i \in J_r} H_i$$

$$l_2 = t(J_r, 2) + \sum_{i \in J_r} H_i$$

Mencari *Lower Bound* (LB) pada node 1, 2, 3 dan 4. Bisa dilihat pada tabel 4, 5, 6 dan 7.

LB 1

l1	13	55	9	77
l2	34	42		76

Tabel 4

LB2

l1	16	52	9	77
l2	36	43		79

Tabel 5

LB3

l1	14	54	13	81
l2	23	54		77

Tabel 6

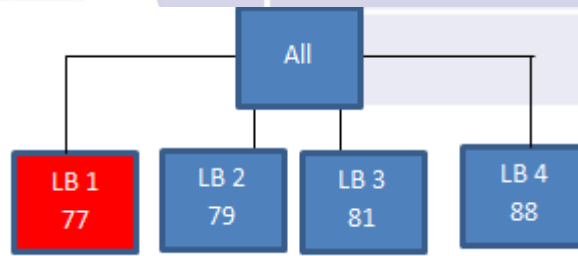
LB 4

l1	25	43	20	88
l2	38	30		68

Tabel 7

Langkah 4 yaitu mencari nilai maksimum( $l_1, l_2$ )

- LB 1 = maksimum(77,76) = 77
- LB 2 = maksimum(77,79) = 79
- LB 3 = maksimum(81,77) = 81
- LB 4 = maksimum(88,68) = 88



Gambar 4

Didapat *makespan* terendah dari LB 1, LB 2, LB 3 dan LB 4, yaitu LB 1 senilai 77 maka ulangi langkah 3 dan 4 dengan melakukan percabangan pada *node* 12, 13 dan 14. Bisa dilihat pada tabel 8, 9 dan 10.

LB 12

l1	29	39	9	77
l2	54	22		76

Tabel 8

LB 13

l1	27	41	13	81
l2	43	33		76

Tabel 9

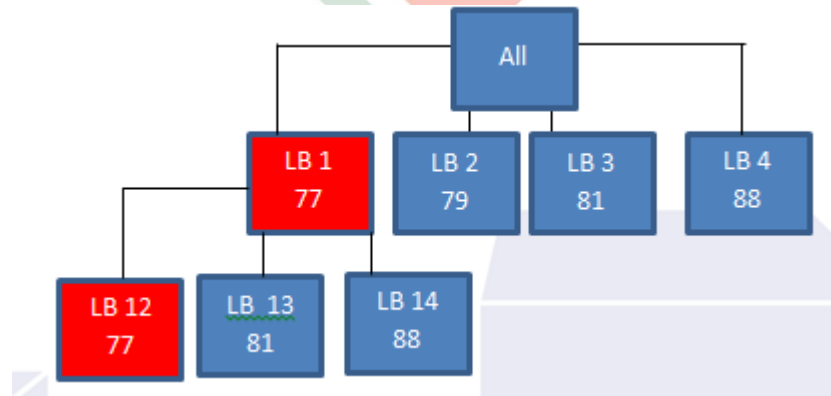
LB 14

l1	38	30	20	88
l2	47	29		76

Tabel 10

Didapatkan nilai maksimum( $l_1, l_2$ ) yaitu :

- LB 12 = maks(77,76) = 77
- LB 13 = maks(81,76) = 81
- LB 14 = maks(88,76) = 88



Gambar 5

Didapat *makespan* terendah dari LB 12, LB 13 dan LB 14, yaitu LB 12 senilai 77 maka ulangi langkah 3 dan 4 dengan melakukan percabangan pada *node* 123 dan 124. Bisa dilihat pada tabel 11 dan 12

LB 123

l1	43	25	13	81
l2	63	13		76

Tabel 11

LB 124

l1	54	14	9	77
l2	67	9		76

Tabel 12

Didapatkan nilai maksimum( $l_1, l_2$ ) yaitu :

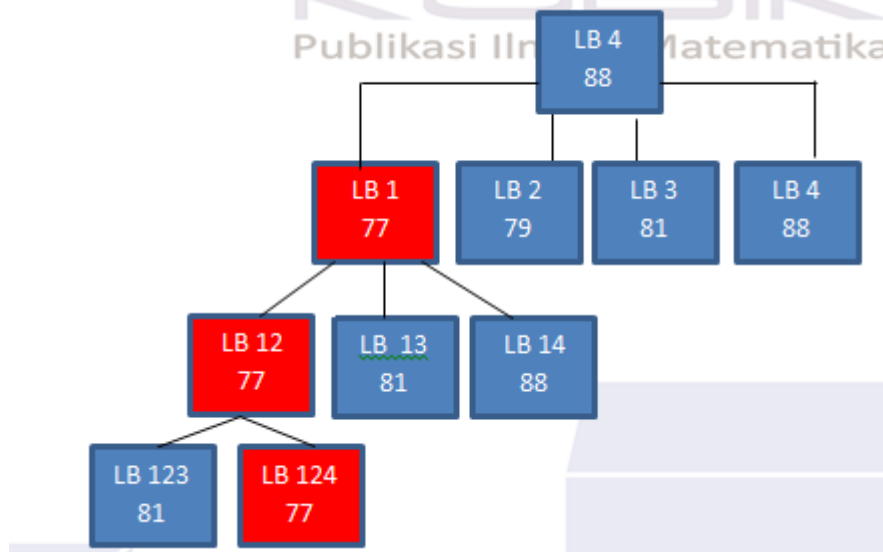
- LB 123 = maks(81,76) = 81
- LB 124 = maks(77,76) = 77

Jadi, didapat nilai maksimum keseluruhan. Bisa dilihat pada tabel 13

Node	LB( $J_r$ )
1	77
2	79
3	81
4	88
12	77
13	81
14	88
123	81
124	77

Tabel 13

Langkah kelima, yaitu membuat penjadwalan yang optimal dengan akar pohon.



Gambar 6

Langkah keenam, yaitu penjadwalan optimal dengan pengurutan *job*, adapun pengurutan *job* yang terjadi adalah 1 , 2, 4 , 3. Dan waktu yang terlewati (*elapsed time*) dapat dicari dengan rumus :

$$\begin{aligned}
 \text{Waktu yang terlewati (elapsed time)} &= \text{makespan terendah} - \sum t_i \\
 &= 77 - 18 \\
 &= 59
 \end{aligned}$$

Sehingga, waktu yang terlewati (*elapsed time*) adalah 59 unit satuan waktu.

**Kesimpulan**

Produksi tipe *flowshop* dua tahap merupakan alur produksi dengan menggunakan dua mesin. Adapun urutan penjadwalan flowshop dengan metode *Branch and Bound* adalah 1, 2,4 dan 3 dengan *makespan* sebesar 77 dan waktu yang terlewati (*elapsed time*) adalah 59 unit satuan waktu.

**Ucapan Terima Kasih**

Penulis mengucapkan terima kasih kepada pihak yang ikut berperan serta dalam penelitian ini, yaitu kepada Dr. Elis Ratna Wulan, S.Si., MT dan dosen lainnya di jurusan matematika Universitas Islam Negeri Sunan Gunung Djati Bandung. Dan juga untuk rekan seperjuangan yang selalu mendukung dalam penelitian ini.

**Referensi**

- [1] Baker, Kenneth. 1974. *Introduction to Sequencing and Scheduling*. America : John Wiley and Son Inc.
- [2] Deepak Gupta, dkk. (2013). *Optimal Two Stage Flow Shop Scheduling Problem with Branch And Bound Method Including Transportation Time*. International Journal of Innovative Research in Computer and Communication Engineering. Maharishi Markandeshwar University, Mullana, Ambala, India.
- [3] Eko Wicaksono, Farouq (dkk.). (2014). *Simulasi Aturan Johnson Untuk Penjadwalan Produksi Flowshop di Perusahaan Furnitur*. Dalam Repositori Jurnal Mahasiswa PTIIK UB. [online] Vol 3 (8).
- [4] Kurniawan Tirtha, Dhany. (2007). *Perancangan Program Aplikasi Penjadwalan Produksi Filling Cabinet M Menggunakan Algoritma Smith Pada Pt. Elite Permai Metal Works Ltd*. Skripsi pada Universitas Bina Nusantara. Jakarta.
- [5] Laila Rizki Fauziah, Nur Rahma. (2013). *Laporan Praktikum Perencanaan Dan Pengendalian Produksi Acara V Penjadwalan Jobshop Dan Flowshop*. Laporan Praktikum pada laporan praktikum sistem produksi Universitas Gajah Mada, Yogyakarta.
- [6] Mercado, R. Z. R. dan Bard, J. F., (1997), *A Branch and Bound Algorithm for Flowshop Scheduling with Setup Times*, University of Texas, Austin.
- [7] Merza Khaliel. (2005). *Usulan Penjadwalan Produksi Pada Proses Produksi Flowshop Dengan Metode CDS Dan Branch And Bound di PT. Kabelindo Murni TBK*. Skripsi pada Universitas Bina Nusantara. Jakarta.