

Pengaruh *Learning Effect* Bergantung-Waktu dan Deteriorasi Terhadap Penjadwalan Mesin Tunggal

Dian Ayu Nurul Ihsani¹, Sunarsih², Robertus Heri³

^{1,2,3}Jurusan Matematika FSM Universitas Diponegoro Semarang
¹dienihsani@gmail.com

Abstrak

Learning effect and deterioration do not fragmentary happen all the time. If both of them simultaneously found, the processing time of the job will increase yet decrease from the plan at once. The actual processing time of jobs are defined by function of their starting times and positions in the sequence. The effect of learning and deterioration on single machine scheduling at this paper is applied at a paper-mill. Learning effect as a result of regular performance-evaluation at this paper-mill reduce the effect of deterioration up to 206,5509 hours. Routing jobs by Earlier Due Date (EDD) rule construct the optimal result under maximum lateness case in this paper than either Most Urgent Job (MUJ) or Shortest Processing Time (SPT) do. The maximum lateness under EDD rule is 13,6% less than sequence that is recently used in that paper-mill.

Key Word: single-machine, deterioration, learning effect, maximum lateness.

1. Pendahuluan

Produk tepat waktu merupakan salah satu kunci perusahaan yang bergerak di bidang manufaktur untuk menghadapi persaingan. Salah satu bagian dari rencana produksi perusahaan untuk mengatasi hal tersebut adalah dengan melakukan penjadwal yang tepat agar seluruh pekerjaan dapat diakomodir oleh sejumlah mesin dan mendapatkan hasil yang optimum. Berhubungan dengan keterlambatan produksi, tugas akhir ini membahas permasalahan *lateness* maksimum pada penjadwalan mesin tunggal yang mengerjakan sejumlah pekerjaan sejenis dengan waktu kedatangan yang sama. Permasalahan *lateness* maksimum dikatakan optimum jika urutan kerja menghasilkan *lateness* maksimum terkecil.

Dalam ruang lingkup mesin tunggal penjadwalan hanya dilakukan untuk mengatur urutan kerja (*sequencing*) karena diasumsikan waktu untuk memproses pekerjaan sudah diketahui. Pada penjadwalan sederhana,

waktu proses dari suatu pekerjaan dianggap bernilai konstan. Padahal pada kenyataannya, perusahaan pasti mengalami beberapa faktor yang menyebabkan waktu proses dari pekerjaan menjadi lebih panjang akibat deteriorasi atau lebih pendek karena *learning effect*.

Dampak dari *learning effect* dan deteriorasi terhadap perubahan waktu proses pekerjaan telah banyak dipelajari. Kuo dan Yang [5] memperkenalkan dampak *learning effect* bergantung pada waktu terhadap penjadwalan mesin tunggal dengan aturan *Shortest Processing Time* (SPT) sebagai solusi optimum. Koulamas dan Kyparisis [4] menunjukkan bahwa $O(n \log n)$ (dengan n adalah jumlah pekerjaan) aturan SPT optimum untuk *makespan* dan meminimalkan *total completion time* pada mesin tunggal jika *learning effect* merupakan fungsi dari jumlah waktu proses pekerjaan yang telah dikerjakan. Browne dan Yechiali [1] menganalisis efek dari deteriorasi terhadap penjadwalan optimum untuk mendapatkan *makespan*

minimal. Cheng, et al. [2] menggunakan algoritma $O(n \log n)$ untuk menyelesaikan permasalahan *due date*, *earliness*, dan *tardiness* pada mesin tunggal yang mengalami deteriorasi linear yang tidak bergantung pada pekerjaan.

Learning effect maupun deteriorasi tidak selalu terjadi secara terpisah. Pada kebanyakan kasus di dunia nyata, keduanya justru lebih sering ditemukan terjadi secara bersamaan. Sun [9] memperkenalkan deteriorasi dan *learning effect* yang terjadi secara bersamaan pada mesin tunggal. Sun menunjukkan bahwa permasalahan *makespan*, *total completion time*, jumlah *quadratic job completion time*, *total weighted completion time*, dan *lateness* maksimum masih memiliki solusi optimum dengan kondisi ini. Yang dan Kuo [12] memperkenalkan dua jenis *learning effect* yang terjadi pada mesin tunggal terdeteriorasi, yaitu *learning effect* bergantung pada pekerjaan dan *learning effect* yang tidak bergantung pekerjaan. Low dan Lin [6] menunjukkan *learning effect* bergantung pada waktu yang terjadi pada ruang lingkup mesin tunggal dan *flowshop* terdeteriorasi memiliki solusi optimum untuk permasalahan *makespan*, *total completion time*, dan *weighted completion time*.

Aplikasi dari model dalam tugas akhir ini disimulasikan pada salah satu pabrik kertas yang menggunakan sebuah mesin tunggal. Deteriorasi terjadi ketika mesin menghasilkan kuantitas *reel* (tonase) kurang dari biasanya atau *gramatur* (g/m^2) kertas yang berbeda sehingga kertas mengalami putus. Perubahan tonase dan *gramatur* ini menyebabkan waktu proses pembuatan kertas meningkat. Untuk mengatasinya, pekerja dituntut untuk menjadi lebih cakap dalam pembuatan bubur kertas dan penyambungan kertas putus sehingga penundaan akibat deteriorasi tidak terlalu tinggi. Kecakapan dari pekerja ini merupakan bentuk *learning effect* yang terjadi. Penjadwalan yang biasa dilakukan di sana adalah dengan mengurutkan pekerjaan berdasarkan bobot. Hal ini menyebabkan beberapa pekerjaan mengalami keterlambatan

sehingga permasalahan *lateness* maksimum dapat diterapkan di pabrik kertas tersebut.

2. Metode Pembahasan

Metode yang digunakan dalam penyusunan tugas akhir ini adalah studi literatur dengan menggunakan jurnal acuan, jurnal pendukung, buku, dan materi penunjang lainnya. Validitas teori di dunia nyata dilakukan dengan menerapkan data produksi berdasarkan observasi, wawancara, dan pengambilan data yang sesuai di suatu pabrik kertas ke dalam model yang digunakan.

3. Hasil dan Pembahasan

3.1. Model Waktu Proses sebenarnya

Mesin tunggal merupakan kasus yang paling sederhana karena operasi dilakukan hanya pada satu mesin. Meski demikian, ruang lingkup mesin tunggal pada prakteknya lebih banyak muncul dibandingkan ruang lingkup mesin lain [3]. Mesin tunggal merupakan kasus khusus dari ruang lingkup mesin yang lain. Penyelesaian permasalahan pada model mesin tunggal biasanya tidak hanya berlaku pada ruang lingkup mesin tunggal, tapi juga bisa menjadi dasar heuristik yang dapat diaplikasikan pada ruang lingkup mesin yang lebih kompleks [8].

Learning effect merupakan salah satu dampak yang didapatkan dari upaya perusahaan untuk meningkatkan daya saing. *Learning effect* bisa diperoleh dari evaluasi terhadap pekerja sebagai sebuah individu maupun sistem produksi perusahaan secara keseluruhan. Seorang pekerja akan menemukan cara untuk melakukan pekerjaan dengan lebih efisien seiring dengan banyaknya pengulangan yang pernah dilakukan untuk melakukan pekerjaan yang sama [5]. Dengan kata lain, *learning effect* akan meningkat seiring dengan

pengalaman yang didapatkan dari pengoperasian suatu pekerjaan. Waktu proses sebenarnya tidak dapat dikalkulasi jika waktu proses yang dijadwalkan dari pekerjaan tersebut tidak diketahui, itulah yang disebut sebagai *learning effect* bergantung-waktu.

Deteriorasi dapat terjadi ketika performa mesin untuk memproses pekerjaan secara berangsur-angsur berkurang. Awalnya mesin dianggap berada pada level performa tertinggi. Penurunan performa mesin tampak pada pekerjaan yang dikerjakan belakangan membutuhkan waktu proses yang semakin panjang [1]. Jika deteriorasi terjadi pada mesin, maka keseluruhan waktu proses akan bertambah akibat keadaan mesin tersebut. Setiap pekerjaan akan mengalami tingkat deteriorasi yang sama karena melewati satu mesin yang sama [2].

Jika *learning effect* dan deteriorasi terjadi secara bersamaan, maka waktu proses sebenarnya dari pekerjaan didefinisikan sebagai fungsi hubungan antara waktu mulai dan posisi urutan dari masing-masing pekerjaan dalam mesin. Low dan Lin [6] memperkenalkan model waktu proses sebenarnya yang dipengaruhi oleh *learning effect* bergantung-waktu dan deteriorasi.

Misalkan ada n pekerjaan yang akan dioperasikan pada sebuah mesin yang hanya bisa mengoperasikan satu pekerjaan pada satu waktu dan tidak diperbolehkan menganggur hingga pekerjaan terakhir selesai dioperasikan, maka waktu proses sebenarnya dari pekerjaan J_j yang dimulai waktu t dan dijadwalkan pada posisi r sebagai:

$$\begin{aligned} P_{jr}(t) &= P_j \left(1 - \frac{\sum_{l=1}^{r-1} P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{r-1} + st \\ &= P_j \left(\frac{\sum_{l=r}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{r-1} + st \end{aligned} \quad (3.1)$$

Di mana $j, r = 1, \dots, n$ dengan

- a : indeks *learning effect* bergantung pada waktu ($a \geq 1$),
- b : indeks *learning effect* bergantung pada posisi ($0 < b < 1$),

- s : tingkat deteriorasi mesin ($s \geq 0$),
- t : waktu pekerjaan dimulai (jam),
- r : posisi pekerjaan dalam urutan,
- $P_{jr}(t)$: waktu proses sebenarnya dari pekerjaan J_j yang dimulai pada waktu t dan dijadwalkan berada di posisi ke- r dalam urutan (jam),
- P_j : waktu proses dari pekerjaan J_j (jam),
- $P_{[l]}$: waktu proses dari pekerjaan J_l yang dijadwalkan berada di posisi ke- l dalam urutan (jam).

Pada model di atas dapat dilihat bahwa waktu proses sebenarnya tidak dapat dikalkulasi jika waktu proses pekerjaan sebelumnya tidak diketahui. Berbeda dengan *learning effect* bergantung pada posisi, pada model di atas r hanya menunjukkan jumlah pekerjaan yang telah dilakukan. Semakin banyak pekerjaan yang pernah dilakukan, maka nilai $\left(\frac{\sum_{l=r}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a$ akan semakin kecil sehingga waktu proses berkurang semakin tajam. Dengan kata lain, proses menjadi lebih efisien sebanding dengan banyaknya pekerjaan yang telah dilakukan [4].

Pekerjaan yang mengalami deteriorasi memiliki tingkat kebergantungan yang tetap terhadap posisi [7]. Persamaan (3.1) tersebut memenuhi prinsip *learning effect* yang sesuai dengan kapasitas proses sebenarnya karena memiliki ketergantungan terhadap posisi akibat deteriorasi, sekaligus ketergantungan terhadap waktu akibat dari *learning effect* itu sendiri [6].

Dalam model tersebut, st menunjukkan hubungan linear antara deteriorasi dengan waktu awal pengerjaan suatu pekerjaan. Secara sederhana st menggambarkan perubahan waktu awal suatu pekerjaan sebanding dengan tingkat deteriorasi yang dialami.

3.2. Waktu Penyelesaian Pekerjaan

Jika diberikan jadwal $\pi = [J_1, J_2, \dots, J_n]$ dengan $P_{[j]}$ sebagai proses normal dari

pekerjaan yang berada dalam urutan ke- j , maka waktu penyelesaian pekerjaan pertama ($j = 1, r = 1$) ketika pekerjaan tersebut berada di posisi awal ($t = 0$) dengan nilai awal $C_{[0]} = 0$ adalah

$$C_{[1]} = 0 + P_{[1]} \left(\frac{\sum_{l=1}^n P_{[1]}}{\sum_{l=1}^n P_l} \right)^a b^0 + s \cdot 0 = P_{[1]}$$

Pekerjaan kedua ($j = 2, r = 2$) dimulai setelah pekerjaan pertama selesai sehingga untuk pekerjaan kedua $t = C_{[1]}$ dan seterusnya. Dengan mensubstitusikan t pada Persamaan (2.1), diperoleh:

$$\begin{aligned} C_{[2]} &= C_{[1]} + P_{[2]} \left(\frac{\sum_{l=2}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{2-1} + s \cdot C_{[1]} \\ &= P_{[1]} + P_{[2]} \left(\frac{\sum_{l=2}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b + s \cdot P_{[1]} \\ &= P_{[1]}(1 + s) + P_{[2]} \left(\frac{\sum_{l=2}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b \end{aligned}$$

Secara umum, waktu penyelesaian pekerjaan ke- k ($j = k$) adalah

$$\begin{aligned} C_{[k]} &= P_{[1]}(1 + s)^{k-1} \\ &+ P_{[2]}(1 + s)^{k-2} \left(\frac{\sum_{l=2}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b \\ &+ P_{[3]}(1 + s)^{k-3} \left(\frac{\sum_{l=3}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^2 + \dots \\ &+ P_{[k]} \left(\frac{\sum_{l=k}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{k-1} \\ &= P_{[1]}(1 + s)^{k-1} \\ &+ \sum_{i=2}^k P_{[i]} (1 + s)^{i-2} \left(\frac{\sum_{l=i}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{i-1} \end{aligned}$$

sedangkan untuk $j = k + 1$ diperoleh

$$\begin{aligned} C_{[k+1]} &= P_{[1]}(1 + s)^{k-1} \\ &+ P_{[2]}(1 + s)^{k-2} \left(\frac{\sum_{l=2}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b \\ &+ P_{[3]}(1 + s)^{k-3} \left(\frac{\sum_{l=3}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^2 \\ &+ \dots + P_{[k]} \left(\frac{\sum_{l=k}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{k-1} \\ &+ P_{[k+1]}(1 + s)^{-1} \left(\frac{\sum_{l=k+1}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^k \end{aligned}$$

$$\begin{aligned} &= P_{[1]}(1 + s)^{k-1} \\ &+ \sum_{i=2}^k P_{[i]} (1 + s)^{i-2} \left(\frac{\sum_{l=i}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{i-1} \\ &+ P_{[k+1]}(1 + s)^{-1} \left(\frac{\sum_{l=k+1}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^k \\ &= P_{[1]}(1 + s)^{k-1} \\ &+ \sum_{i=2}^{k+1} P_{[i]} (1 + s)^{i-2} \left(\frac{\sum_{l=i}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{i-1} \end{aligned}$$

Karena untuk $j = k + 1$ benar, maka dengan menggunakan induksi matematika, persamaan tersebut berlaku untuk seluruh k bilangan asli. Waktu penyelesaian untuk seluruh pekerjaan ($j = n$) dapat ditulis sebagai

$$C_{[n]} = P_{[1]}(1 + s)^{n-1} + \sum_{i=2}^n P_{[i]} (1 + s)^{i-2} \left(\frac{\sum_{l=i}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{i-1} \quad (3.2)$$

$$3.3. \text{ Permasalahan 1 } \left| P_j \left(\frac{\sum_{l=r}^n P_{[l]}}{\sum_{l=1}^n P_l} \right)^a b^{r-1} + st \right| L_{max}$$

Untuk permasalahan *lateness* dalam penjadwalan mesin tunggal, ditunjukkan

bahwa solusi optimum diperoleh berdasarkan Teorema 3.1 dan Akibat 3.1 di bawah ini:

Teorema 3.1

Pada permasalahan $1|P_j(\alpha(t) + \beta r^a)|L_{max}$, jika pekerjaan-pekerjaannya memiliki bobot yang disetujui ($P_i \leq P_j$) yang secara tidak langsung berarti $d_i \leq d_j$, untuk setiap J_i dan J_j , jadwal optimum didapatkan dengan mengurutkan pekerjaan dalam order d_j tak turun (aturan EDD) [10].

Bukti

Misalkan S_1 adalah jadwal dari pekerjaan J_i dan J_j yang berdekatan yang tidak menggunakan aturan EDD, dengan kata lain $d_i > d_j$ yang berarti $P_i \geq P_j$ sehingga $S_1 = \{\pi_1, J_i, J_j, \pi_2\}$. Dibuat jadwal lain dengan menukar urutan pekerjaan- i dengan pekerjaan- j . π_1 dan π_2 merupakan suatu urutan parsial yang boleh kosong. Dapat dilihat bahwa penukaran urutan akan menciptakan jadwal baru $S_2 = \{\pi_1, J_j, J_i, \pi_2\}$. Diasumsikan ada sejumlah $(r - 1)$ pekerjaan di π_1 dengan waktu penyelesaian dari pekerjaan terakhirnya adalah B . Jadi, J_i dan J_j merupakan urutan ke- r dan $(r + 1)$ dalam S_1 . Dengan asumsi yang sama, J_j dan J_i merupakan urutan ke- r dan $(r + 1)$ dalam S_2 .

Waktu penyelesaian pekerjaan pada S_1 sesuai dengan kondisi $P_{j_r}(t) = P_j(\alpha(t) + \beta r^a)$ seperti pada Teorema 3.1. di atas adalah:

$$C_i(S_1) = B + P_i(\alpha B + \beta r^a)$$

$$C_j(S_1) = B + \beta P_i r^a + P_i \alpha B + P_j \alpha B + P_i(\alpha B + \beta r^a) + \beta P_j(r + 1)^a$$

sedangkan waktu penyelesaian pada S_2 :

$$C_j(S_2) = B + P_j(\alpha B + \beta r^a)$$

$$C_i(S_2) = B + \beta P_j r^a + P_j \alpha B + P_i \alpha B + P_j(\alpha B + \beta r^a) + \beta P_i(r + 1)^a$$

Sehingga, *lateness* menjadi:

$$L_i(S_1) = B + P_i(\alpha B + \beta r^a) - d_i$$

$$L_j(S_1) = B + \beta P_i r^a + P_i \alpha B + P_j \alpha B + P_i(\alpha B + \beta r^a) + \beta P_j(r + 1)^a - d_j$$

$$L_j(S_2) = B + P_j(\alpha B + \beta r^a) - d_j$$

$$L_i(S_2) = B + \beta P_j r^a + P_j \alpha B + P_i \alpha B + P_j(\alpha B + \beta r^a) + \beta P_i(r + 1)^a - d_i$$

Misalkan $L(S_1)$ adalah *lateness* dari pekerjaan ke- $(r + 2)$ pada S_1 dan $L(S_2)$ adalah *lateness* dari pekerjaan ke- $(r + 2)$ pada S_2 , dapat dilihat bahwa $L(S_1) = L(S_2)$. Ambil $L_i(S_1)$ dan $L_k(S_1)$ adalah *lateness* J_i dan J_k pada S_1 sementara $L_i(S_2)$ dan $L_k(S_2)$ adalah *lateness* J_i dan J_k pada S_2 . Dapat dinyatakan *lateness* maksimum dalam S_1 adalah

$$L_{max}(S_1) = \max\{L(S_1), L_i(S_1), L_k(S_1)\}$$

dan *lateness* maksimum dalam S_2 adalah

$$L_{max}(S_2) = \max\{L(S_2), L_i(S_2), L_k(S_2)\}$$

Karena nilai P_j pasti positif, maka $L_j(S_1) > L_j(S_2)$. Sementara itu dengan $d_i > d_j$ dan $P_i \geq P_j$, maka $L_j(S_1) > L_i(S_2)$. Oleh karena itu diperoleh

$$L_k(S_1) > \max\{L_i(S_2), L_k(S_2)\}$$

$$\max\{L(S_1), L_i(S_1), L_k(S_1)\} > \max\{L(S_1), L_i(S_2), L_k(S_2)\}$$

dengan $L(S_1) = L(S_2)$

$$\max\{L(S_1), L_i(S_1), L_k(S_1)\} > \max\{L(S_2), L_i(S_2), L_k(S_2)\}$$

$$L_{max}(S_1) > L_{max}(S_2)$$

$L_{max}(S_1) > L_{max}(S_2)$ menunjukkan bahwa S_1 bukan jadwal optimum karena ada

jadwal lain yang memiliki *lateness* maksimum lebih kecil. Dari penjelasan tersebut terbukti bahwa untuk meminimalkan *lateness* maksimum, jadwal harus diurutkan sedemikian hingga d_j tak turun (aturan EDD).

Akibat 3.1

Pada permasalahan $1 \left| P_j \left(\frac{\sum_{l=r}^n P_{[l]} \right)^a b^{r-1} + st \right| L_{max}$, jika pekerjaan-pekerjaannya memiliki bobot yang disetujui ($P_i \leq P_j$), yang secara tidak langsung berarti $d_i \leq d_j$, untuk setiap J_i dan J_j , jadwal optimum bisa didapatkan dengan mengurutkan pekerjaan dalam order d_j tak turun (aturan EDD).

3.4. Simulasi Kasus di Industri Manufaktur

Suatu pabrik kertas yang mengalami tingkat deteriorasi $s = 0,004$ memperkirakan dapat memproduksi 75 ton kertas perhari (2.250 ton perbulan). Berdasarkan data produksi periode Mei 2014, ternyata kertas yang dihasilkan dalam satu bulan hanya 2.188 ton. Dari waktu proses 159 pekerjaan periode Bulan Mei dari data yang diperoleh, tingkat *learning effect* $a = 1,4$ dan $b = 0,99$.

Waktu penyelesaian normal berdasarkan data adalah 700.16 jam. Meski bersifat kualitatif, berdasarkan data periode Mei 2014, hasil dari *learning effect* yang terjadi di perusahaan tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1. Perbandingan Waktu Penyelesaian (jam) Pekerjaan

Normal	$s = 0.004$	$s = 0.004,$ $a = 1.4, b = 0.99$
700.16	926.4909	719.94

Meski masih menyebabkan kemunduran 19.76 jam, namun *learning effect* yang ada di pabrik kertas tersebut telah meredam deteriorasi hingga 206.5509 jam.

Penjadwalan produksi yang biasa dilakukan di pabrik kertas tersebut adalah

mengurutkan pekerjaan berdasarkan kepentingan pekerjaan. Urutan ini berdasarkan aturan penjadwalan Donald Waters [11], merupakan urutan *Most Urgent Job First* (MUJ). Selanjutnya untuk mengetahui urutan penjadwalan yang tepat dalam permasalahan *lateness* maksimum, diadakan perbandingan antara hasil urutan yang biasa dilakukan di pabrik kertas tersebut (MUJ), *Shortest Processing Time* (SPT), dan *Earlier Due Date* (SPT). Hasil dari perbandingan ini dapat dilihat pada Tabel 3.2.

Tabel 3.2. Perbandingan Hasil Urutan dengan Aturan MUJ, SPT, dan EDD

	MUJ	SPT	EDD
Barang terlambatan (unit)	65	59	59
L_{max} (jam)	1859,2	1735,678	1606,856

Pada Tabel 3.2 dapat dilihat bahwa aturan EDD memberikan *lateness* maksimum terkecil. Jika dibandingkan dengan urutan yang saat ini dilakukan di pabrik kertas tersebut, aturan EDD menurunkan *lateness* maksimum sebesar 13.6%.

4. Kesimpulan

Model waktu proses sebenarnya pada mesin tunggal dengan *learning effect* bergantung pada waktu dan deteriorasi dapat diterapkan dalam urutan penjadwalan produksi dengan aturan *Earliest Due Date* (EDD). Dalam simulasi kasus di suatu pabrik kertas, urutan dengan aturan EDD menghasilkan *lateness* maksimum yang lebih kecil dibandingkan urutan dengan aturan MUJ dan aturan SPT. *Lateness* maksimum dari penjadwalan produksi dengan aturan EDD mengalami penurunan sebesar 13,6% jika dibandingkan dengan urutan yang selama ini digunakan di pabrik kertas tersebut.

5. Referensi

- [1] Browne, S. Yechiali, U. 1990. Scheduling Deteriorating Jobs on a Single Processor. *Operations Research* Vol 38 PP 495-498.
- [2] Cheng, T C E. Kang, L. Ng, C T. 2004. Due-date assignment and single-machine scheduling with deteriorating jobs. *Journal of the Operational Research Society* Vol 55 PP 198-203.
- [3] French, S. 1982. Sequencing & Scheduling: An Introduction to the Mathematics of the Job-Shop. England: ELLIS HORWOOD LIMITED.
- [1] Koulamas, C. Kyparisis, G J. 2007. Single-Machine and Two-Machine Flowshop Scheduling with General Learning Functions. *European Journal of Operational Research* Vol 178 PP 402-407.
- [2] Kuo, W H. Yang, D L. 2006. Minimizing the Total Completion Time in a Single-Machine Scheduling Problem with a Time-Dependent Learning Effect. *European Journal of Research* Vol 174 PP 1184-1190.
- [3] Low, C. Lin, W Y. 2013. Some Scheduling Problems with Time-Dependent Learning Effect and Deteriorating Jobs. *Applied Mathematical Modelling* Vol 37 PP 8865-8875.
- [4] Mosheiov, G. 1994. Scheduling jobs under simple linear deterioration. *Computer & Operations Research* Vol 21 PP 653-659.
- [5] Pinedo, M. 2002. Scheduling Theory, Algorithms, & System. New Jersey: PRENTICE-HALL, Inc.
- [6] Sun, L. 2009. Single-Machine Scheduling Problems with Deteriorating Jobs and Learning Effects. *Computers & Industrial Engineering* Vol 57 PP 843-846.
- [7] Wang, J B. 2007. Single-machine scheduling with the effects of learning & deteriorating. *Omega* Vol 35 PP 397-402.
- [8] Waters, D. 2003. Logistics an Introduction to Supply Chain Management. New York: PALGRAVE MACMILLAN.
- [9] Yang, D L. Kuo, W H. 2010. Some Scheduling Problems with Deteriorating Jobs and Learning Effects. *Computers & Industrial Engineering* Vol 58 PP 25-28.