

PENJADWALAN *FLOW SHOP N JOB M* MESIN DENGAN METODE *FIRST COME FIRST SERVED (FCFS)*, *EARLIEST DUE DATE (EDD)* DAN ALGORITMA HEURISTIK POUR

Dwi Agustina Kurniawati, Abdul Latief Irsyad
Program Studi Teknik Industri, Fakultas Teknik, Universitas Islam Negeri Sunan Kalijaga
dwi.kurniawati@uin-suka.ac.id

Abstrak

Penelitian ini membahas tentang penjadwalan flow shop n job m mesin dengan pendekatan metode First Come First Served (FCFS), Earliest Due Date (EDD) dan Algoritma Heuristik Pour. Tujuan penelitian ini adalah melakukan perbandingan performansi antara ketiga metode tersebut untuk meminimalkan makespan. Dalam penelitian ini data waktu proses dan due date yang digunakan merupakan data random yang dibangkitkan dengan software Matlab 7.1. Kombinasi jumlah job dan mesin yaitu 4 job dan 2 mesin. Penelitian ini dilakukan dalam dua tahap. Tahap pertama yaitu melakukan perhitungan menggunakan metode enumerasi. Tahap kedua yaitu melakukan perhitungan dengan metode FCFS, EDD dan Pour. Parameter performansi yang digunakan yaitu makespan rata-rata, Efficiency Index (EI), Relative Error (RE), run time, dan nilai deviasi makespan rata-rata. Hasil dari penelitian ini menunjukkan bahwa metode terbaik untuk penjadwalan flow shop n job m mesin dalam meminimalkan makespan adalah Algoritma Heuristik Pour karena memiliki nilai makespan rata-rata terkecil yaitu 30,70 time unit, nilai EI < 1, nilai RE lebih dari 5% dan memiliki nilai deviasi yang terkecil yaitu 0,35 time unit. Meskipun metode Algoritma Heuristik Pour memiliki nilai run time yang paling lama akan tetapi nilai run time yang diperoleh masih dapat ditoleransi yaitu sebesar 2.741351 detik.

Kata kunci : Penjadwalan, *flow shop*, enumerasi, *FCFS*, *EDD*, *Pour*, makespan rata-rata, efficiency index, relative error, run time.

I. PENDAHULUAN

Proses produksi merupakan serangkaian proses untuk menghasilkan barang atau jasa. Pada proses produksi inilah terdapat berbagai macam kegiatan seperti proses perencanaan produksi, penentuan kualitas, penjadwalan kegiatan produksi, sampai pengolahan limbah yang dihasilkan selama proses produksi berlangsung. Semua proses produksi tersebut sangat penting, salah satunya adalah penjadwalan kegiatan produksi.

Menurut Ginting (2009), penjadwalan adalah pengurutan pembuatan atau pengerjaan produk secara menyeluruh yang dikerjakan pada beberapa buah mesin. Menurut Baker (2009), penjadwalan *flow shop* adalah salah satu jenis penjadwalan produksi dimana setiap *n job* akan melalui setiap *m* mesin dengan urutan yang seragam.

Masalah penjadwalan yang sering muncul yaitu terjadi kesulitan dalam menentukan *job* mana yang harus dikerjakan terlebih dahulu agar mendapatkan hasil yang optimal. Metode atau cara untuk menentukan *job* mana yang harus dikerjakan terlebih dahulu dalam penjadwalan *flow shop* adalah dengan aturan prioritas. Menurut Bedworth (1987) dalam Rasjidin dan Hidayat (2006) metode-metode yang digunakan untuk menentukan *job* mana yang dikerjakan terlebih dahulu, yaitu : *Short Processing Time (SPT)*, *Weighted Short Processing Time (WSPT)*, *First Come First Served (FCFS)*, *Earliest Due Date (EDD)*, *Slack Time (ST)*, *Longest Processing Time (LPT)*, *Random Selection (RS)*, *Heighest Expected Profitability (HEP)*, *Prefered Customer Order (PCO)*, dan Algoritma Hodgson.

Dalam perkembangan selanjutnya banyak teknik penjadwalan dengan menggunakan metode heuristik dan metaheuristik. Beberapa metode heuristik yang telah digunakan untuk menyelesaikan permasalahan penjadwalan *flow shop* adalah Algoritma Pour, Algoritma Campbell, Dudek dan Smith (CDS), Algoritma Palmer, Algoritma Nawaz, Ensore dan Ham (NEH), Algoritma Gupta, Algoritma Dannenbring. Adapun metode metaheuristik yang telah digunakan adalah *Tabu Search*, *Genetic Algorithm*, *Simulated Annealing* dll.

Penelitian Soetanto dan Palit (2004) menunjukkan bahwa algoritma heuristik Pour memberikan *performansi* yang cukup baik dalam menyelesaikan permasalahan penjadwalan *flow shop* untuk meminimalkan *makespan* dibandingkan dengan metode optimasi *Mixed Integer Programming* (MIP). Penelitian yang dilakukan oleh Kuncoro (2013) menunjukkan bahwa algoritma heuristik Pour memiliki *performansi* yang sama dengan algoritma Nawaz, Ensore dan Ham (NEH) dengan *makespan* minimum dan kompleksitas waktu yang sama.

Banyak peneliti yang telah melakukan *computational study* atau studi komputasi untuk menentukan *performansi* masing-masing metode penjadwalan dan untuk membandingkan *performansi* dari beberapa metode penjadwalan tersebut. Dari *computational study* yang telah dilakukan akan didapatkan metode penjadwalan terbaik sesuai dengan parameter yang telah ditentukan diawal penelitian, misalnya *makespan* terkecil dan *lateness* terkecil.

Arisha et al. (2002) melakukan penelitian dengan program simulasi untuk menemukan metode penjadwalan yang optimum/mendekati optimum dengan tujuan untuk meminimalkan *makespan*. Dalam penelitian tersebut dilakukan perbandingan *performansi* dari lima aturan prioritas yaitu FCFS, SPT, LPT, SRPT dan LRPT. Data yang digunakan dalam penelitian tersebut merupakan data random yang dibangkitkan menggunakan *software* pada komputer.

Berdasarkan hal-hal di atas, maka akan dilakukan *computational study* (studi komputasi) untuk membandingkan *performansi* dari metode FCFS, EDD dan Pour dalam menentukan penjadwalan *flow shop* n job m mesin. Penelitian ini bertujuan untuk meminimalkan *makespan*, dimana data yang digunakan dalam studi komputasi ini adalah data random yang dibangkitkan oleh komputer.

II. DEFINISI DALAM PENJADWALAN

Ginting (2009) dalam bukunya Penjadwalan Mesin menjabarkan beberapa definisi yang digunakan dalam penjadwalan mesin :

1. *Processing time* (t_i)

Adalah waktu yang dibutuhkan untuk mengerjakan suatu pekerjaan. Dalam waktu proses ini sudah termasuk waktu yang dibutuhkan untuk persiapan dan pengaturan (*set-up*) selama proses berlangsung.

- *Processing time*, taksiran peramalan tentang berapa lama waktu yang dibutuhkan untuk menyelesaikan suatu tugas. Taksiran meliputi *setup time* yang mungkin dibutuhkan, yang diasumsikan bebas. *Processing time* untuk tugas i dinyatakan dengan t_i .

2. *Due date* (d_i)

Adalah batas waktu dimana operasi terakhir dari suatu pekerjaan harus selesai.

- *Due date*, batas waktu yang ditentukan untuk tugas yang telah lewat, yang akan dinyatakan dengan terlambat. Diasumsikan bahwa akan diberikan denda bila terlambat. *Due date* dinyatakan dengan d_i .

3. *Slack time* (SL_i)

Adalah waktu tersisa yang muncul akibat dari waktu prosesnya lebih kecil dari *due date* nya.

$$SL_i = d_i - t_i$$

- *Slack*, ukuran perbedaan waktu antara waktu sisa dari batas waktu tugas dengan waktu prosesnya (*processing time*). *Slack* dinyatakan dengan SL_i : $SL_i = d_i - t_i$.

4. *Flow time* (F_i)

Flow time, rentang waktu antara satu titik dimana tugas tersedia untuk diproses dengan suatu titik ketika tugas tersebut selesai. Jadi, *flow time* sama dengan *processing time* dijumlahkan dengan waktu ketika tugas menunggu sebelum diproses. *Flow time* dinyatakan dengan F_i .

5. *Completion time* (C_i)

Adalah waktu yang dibutuhkan untuk menyelesaikan pekerjaan mulai dari saat tersedianya pekerjaan ($t=0$) sampai pada pekerjaan tersebut selesai dikerjakan.

- *Completion time*, rentang antara awal dari tugas pada pekerjaan pertama, dimana waktunya mengacu pada $t=0$, dengan waktu ketika tugas selesai. Simbol dinyatakan dengan C_i .
- 6. *Lateness* (L_i)
Adalah selisih antara *Completion time* (C_i) dengan *due date* nya (d_i). Suatu pekerjaan memiliki *lateness* yang bernilai positif apabila pekerjaan tersebut diselesaikan setelah *due date* nya, pekerjaan tersebut akan memiliki keterlambatan yang negatif. Sebaliknya jika pekerjaan diselesaikan setelah batas waktunya, pekerjaan tersebut memiliki keterlambatan yang positif.
- 7. *Tardiness* (T_i)
Adalah ukuran waktu terlambat yang bernilai positif jika suatu pekerjaan dapat diselesaikan lebih cepat dari *due date* nya, pekerjaan tersebut akan memiliki keterlambatan yang negatif. Sebaliknya jika pekerjaan diselesaikan setelah batas waktunya, pekerjaan tersebut memiliki keterlambatan yang positif.
 - *Tardiness*, ukuran dari *lateness* positif. Jika tugasnya selesai cepat, maka akan memiliki *lateness* negatif tetapi *tardiness*=0. Jika tugas memiliki *lateness* positif, maka akan memiliki *tardiness* positif juga. *Tardiness*, dinyatakan dengan T_i , dimana T_i adalah maksimum dari $\{0, L_i\}$.
- 8. *Makespan* (M)
Adalah total waktu penyelesaian pekerjaan-pekerjaan mulai dari urutan pertama yang dikerjakan pada mesin atau *work center* pertama sampai kepada urutan pekerjaan terakhir pada mesin atau *work center* terakhir.
- 9. *Heuristic*
Prosedur penyelesaian suatu masalah atau aturan ibu jari (*rule of thumb*) yang ditunjukkan untuk memperoleh hasil yang baik tetapi tidak menjamin hasil yang optimal.

A. KRITERIA DALAM PENJADWALAN

Didalam pengambilan keputusan tentang penjadwalan banyak kriteria yang ditampilkan sebagai evaluasi dari penjadwalan sejumlah *job* yang diproses di dalam sejumlah mesin yang merupakan fungsi dari sekumpulan waktu penyelesaian (Ginting, 2009). Misalkan ada sebanyak n *job* yang akan dijadwalkan, maka kriteria penjadwalan dapat berupa :

- Minimasi waktu penyelesaian rata-rata (*mean flow time*)

$$F = \frac{1}{n} \sum_{j=1}^n F_j \text{ dimana } F_j = t_j + W_j$$

- Minimasi *mean lateness*

$$L = \frac{1}{n} \sum_{j=1}^n L_j$$

- Minimasi *mean tardiness*

$$T = \frac{1}{n} \sum_{j=1}^n T_j \text{ dimana } T_j = \max(0, L_j)$$

- Maksimasi *tardiness*

$$T_{\max} = 1 \leq j \leq n^{(T_j)}$$

- Maksimasi keseluruhan waktu penyelesaian *job* yang ada (*maximum flow time*)

$$F_{\max} = 1 \leq j \leq n^{(F_j)}$$

- Minimasi jumlah *job* yang terlambat (*number of tardy job*)

$$N_T = \sum_{j=1}^n \partial(T_j) \text{ dimana } \partial(T_j) = 1 \text{ jika } T_j > 0$$

$$(T_j) = 0 \text{ jika } T_j \leq 0$$

- Maksimasi utilitas rata-rata mesin (U)

$$U = \frac{\sum_{j=1}^n U_m}{m} \text{ dimana } U_m = \frac{\sum_{j=1}^n t_j}{F_{max}}$$

B. JENIS-JENIS PENJADWALAN

Penjadwalan terbagi menjadi beberapa jenis berdasarkan sistem produksi yang akan dibuat penjadwalannya antara lain penjadwalan *flow shop* dan *job shop* yang masing-masing memiliki kriteria tersendiri. (Ginting, 2009)

a. Penjadwalan *Flow Shop*

Penjadwalan *flow shop* menurut Baker (1974) dalam (Ginting, 2009), merupakan suatu pergerakan unit-unit yang terus-menerus melalui suatu rangkaian stasiun-stasiun kerja yang disusun berdasarkan produk. Susunan suatu proses produksi jenis *flow shop* dapat diterapkan dengan tepat untuk produk-produk dengan desain yang stabil dan diproduksi secara banyak (volume produk), sehingga investasi dengan tujuan khusus (*special purpose*) yang dapat secepatnya kembali.

Suatu masalah kritis dalam *flow shop* adalah pengelompokkan tugas-tugas yang dibutuhkan dalam stasiun kerja, sehingga dicapai suatu kondisi yang memenuhi pembatas-pembatas urutan dan terjadi keseimbangan pada tingkat *output* produksi. Jika tingkat *output* bervariasi untuk masing-masing stasiun kerja, maka hal ini berarti bahwa lintasan produksi tersebut tidak seimbang. Ketidakseimbangan lintasan akan menghasilkan aliran yang tidak teratur dan rendahnya utilitas kapasitas yang disebabkan turunnya kecepatan aliran pada stasiun-stasiun penyebab *bottleneck*. *Flow shop* adalah pengerjaan unit-unit yang terus menerus melalui suatu rangkaian. Stasiun-stasiun kerja yang disusun berdasarkan produk.

Masalah yang kritis pada *flow shop* :

1. Pengelompokkan tugas-tugas yang dibutuhkan dalam stasiun kerja sehingga dicapai kesetimbangan pada tingkat *output* dan memenuhi pembatasan urutan.
2. Ketegangan yang diakibatkan susunan aliran lini terhadap pekerja. Pekerja akan bosan karena terbatasnya variasi kerja pada tiap stasiun dan panjang rentang pengendalian sepanjang lintasannya.

Prioritas *order* pada *flow shop* dipengaruhi terutama pada pengirimnya dibandingkan tanggal pemrosesan. Dengan syarat, *flow shop* digunakan khusus hanya untuk satu jenis produk.

b. Penjadwalan *Job Shop*

Penjadwalan pada proses produksi tipe *job shop* lebih sulit dibandingkan penjadwalan *flow shop*. Hal ini disebabkan oleh 3 alasan yaitu (Ginting, 2009)

1. *Job shop* menangani variasi produk yang sangat banyak, dengan pola aliran yang berbeda-beda melalui pusat-pusat kerja.
2. Peralatan pada *job shop* digunakan bersama-sama oleh bermacam-macam *order* pada prosesnya, sedangkan peralatan pada *flow shop* digunakan khusus untuk satu jenis produk.
3. *Job* yang berbeda mungkin ditentukan oleh prioritas berbeda pula. Hal ini mengakibatkan produk tertentu yang dipilih harus diproses seketika pada saat order tersebut ditugaskan pada suatu pusat kerja. Sedangkan pada *flow shop* tidak terjadi permasalahan seperti tersebut karena keseragaman *output* yang diproduksi untuk persediaan. Prioritas *order* pada *flow shop* dipengaruhi terutama pada pengirimannya dibandingkan tanggal pemrosesan.

Faktor-faktor tersebut menghasilkan sangat banyak kemungkinan kombinasi dari pembebanan (*loading*) dan urutan-urutan (*sequencing*). Perhitungan dari identifikasi dan evaluasi jadwal-jadwal yang mungkin menjadi sulit sehingga banyak perhatian diarahkan pada riset penjadwalan *job shop*. Selain itu, persiapan suatu

penjadwalan *job shop*, penyesuaian dan pembaharuannya membutuhkan investasi yang besar untuk fasilitas komputer.

Pada bagian ini akan dibahas masalah penjadwalan *job shop* dengan memperhatikan permasalahan pada *job loading* dan *job sequencing*. Adapun masalah *job shop* tersebut sebagai berikut :

1. *Job shop loading* artinya memutuskan pusat-pusat kerja yang mana suatu *job* harus ditugaskan. Menggunakan *Gantt chart* dan metode penugasan.
2. *Job sequencing* artinya kita harus menentukan bagaimana urutan proses dari bermacam-macam *job* harus ditugaskan pada mesin-mesin tertentu atau pusat kerja tertentu.

C. PENJADWALAN PRODUKSI *FIRST COME FIRST SERVED* (FCFS)

FCFS (*First Come First Served*), pekerjaan diproses berdasarkan urutan kedatangannya. Yang pertama datang, yang pertama dilayani. Pekerjaan yang pertama datang di sebuah pusat kerja diproses terlebih dahulu (Heizer dan Render, 2010).

D. PENJADWALAN PRODUKSI *EARLIEST DUE DATE* (EDD)

Aturan ini menyebutkan bahwa pengurutan pekerjaan berdasarkan batas waktu (*due date*) tercepat. Pekerjaan dengan saat jatuh tempo paling awal harus dijadwalkan terlebih dahulu dari pada pekerjaan dengan saat jatuh tempo belakangan. Aturan ini bertujuan untuk meminimasi kelambatan maksimum (*maximum lateness*) atau meminimasi ukuran kelambatan maksimum (*maximum tardiness*) suatu pekerjaan. Buruknya, aturan ini akan menyebabkan jumlah pekerjaan yang terlambat menjadi banyak serta akan menambah keterlambatan rata-rata (*mean tardiness*) (Kusuma, 2009).

Langkah-langkah penjadwalan dengan EDD (Baker et al., 2009) :

- a. Mulai dengan urutan yang tidak memperhatikan aturan EDD
- b. Perhatikan pasangan *job i* dan *j* yang berdekatan, dengan *j* mengikuti *i*, atau $d_i > d_j$.
- c. Tukarkan *job i* dan *job j*.
- d. Kembali ke langkah b, sampai urutan EDD terpenuhi.

E. PENJADWALAN PRODUKSI ALGORITMA HEURISTIK POUR

Pour (2001) mengembangkan algoritma heuristik baru didalam menyelesaikan penjadwalan *flow shop* dengan tujuan meminimalkan *makespan* yaitu berdasarkan pendekatan kombinasi. Hal ini dilakukan dengan cara mengganti setiap *job* dengan *job* yang lainnya dalam urutan sampai ditemukan kombinasi urutan yang dapat memenuhi kriteria tujuan.

Dalam metode ini diasumsikan bahwa semua *job* diproses secara terpisah dan *independent* untuk setiap mesinnya. Berikut adalah notasi yang digunakan :

- P_{ij} = waktu proses dari *job i* pada mesin *j*.
- C_{ij} = rentang waktu antara saat *job i* pada mesin *j* dimulai ($t=0$) sampai *job* itu selesai.
- C_i = *sum of completion time* untuk *job i* pada semua mesin.
- F_{max} = rentang waktu antara saat pekerjaan tersedia atau dapat dimulai sampai pekerjaan itu selesai (*makespan*).

Langkah-langkah pengerjaan Algoritma Heuristik Pour :

1. Memilih *job* secara acak sebagai urutan pertama sementara dalam urutan pengerjaan.
2. Menempatkan *job-job* lain (selain *job* yang sudah dipilih sebagai urutan pertama) pada urutan berikutnya.
3. Memilih waktu proses terkecil untuk masing-masing mesin.
4. Melakukan penambahan waktu proses secara *increasing time* pada P_{ij} yang lain, selain P_{ij} paling minimal yang terpilih sebelumnya.
5. Menghitung *sum of completion time* (C_i) untuk setiap *job* yang ada.
6. Mengurutkan C_i dengan aturan *increasing order* untuk diletakkan pada urutan setelah *job* yang sudah dipilih untuk urutan pertama sementara.

7. Setelah didapatkan urutan sementara, maka hitunglah F_{\max} -nya.
8. Melakukan ulang langkah 1-7 untuk setiap *job* yang ada sampai didapatkan F_{\max} paling minimal, yang akan ditempatkan sebagai urutan pertama dari urutan *job*.
9. Melakukan ulang langkah 1-8 sampai semua *job* berada pada urutan pengerjaan.
10. Menghitung *makespan* dari urutan pengerjaan yang telah diperoleh.

F. METODE ENUMERASI

Proses enumerasi, atau pencacahan pengaturan yang memungkinkan dari sekumpulan objek merupakan cara untuk mendapatkan jumlah pengaturan yang mungkin dibuat dari sekumpulan objek tersebut. Jika ada n obyek maka akan ada $n!$ jumlah pengaturan (Riska, 2013).

1. PARAMETER PERFORMANSI

Ada beberapa *performance parameter* yang digunakan untuk menentukan metode yang lebih baik yaitu (Ginting, 2009) :

a. *Efficiency Index* (EI)

Efficiency Index (EI) adalah perbandingan antara algoritma heuristik baru (Pour, 2001) dengan metode yang digunakan perusahaan, dirumuskan sebagai berikut :

$$EI = \frac{F_{\max \text{ proposed algorithm}}}{F_{\max \text{ other algoritm}}}$$

Bila $EI = 1$ maka kedua metode memiliki *performance* yang sama, bila $EI < 1$ maka algoritma heuristik baru memiliki *performance* yang lebih baik dibanding dengan metode yang dibandingkan, demikian juga sebaliknya.

b. *Relative Error* (RE)

Relative Error (RE) digunakan untuk mengetahui seberapa jauh perbedaan kedua metode, yang dapat dihitung sebagai berikut :

$$RE = \frac{F_{\max \text{ other algorithm}} - F_{\max \text{ proposed algorithm}}}{F_{\max \text{ proposed algorithm}}} \times 100\%$$

III. METODOLOGI PENELITIAN

Jumlah *job* dan jumlah mesin yang digunakan untuk penelitian ini yaitu 4 *job* dan 2 mesin (ukuran kecil). Hal ini sesuai dengan tabel kombinasi *job* dan mesin dengan proses random dalam Pour (2001).

Data-data waktu proses dan data *due date* yang akan digunakan untuk studi komputasi dibangun secara random dengan distribusi *uniform*. Pemilihan distribusi *uniform* untuk membangkitkan data-data random karena ingin memunculkan bahwa setiap nilai yang ada diantara interval batas bawah dan batas atas memiliki peluang yang sama untuk muncul. Data-data random tersebut dibangkitkan dengan menggunakan *software* Matlab 7.1.

Data waktu proses memiliki interval $U(1,10)$ dan data *due date* memiliki interval $U(11,20)$. Pemilihan interval $U(1,10)$ pada data waktu proses sesuai dengan Arisha et al. (2002) yang menggunakan interval tersebut dalam penelitiannya.

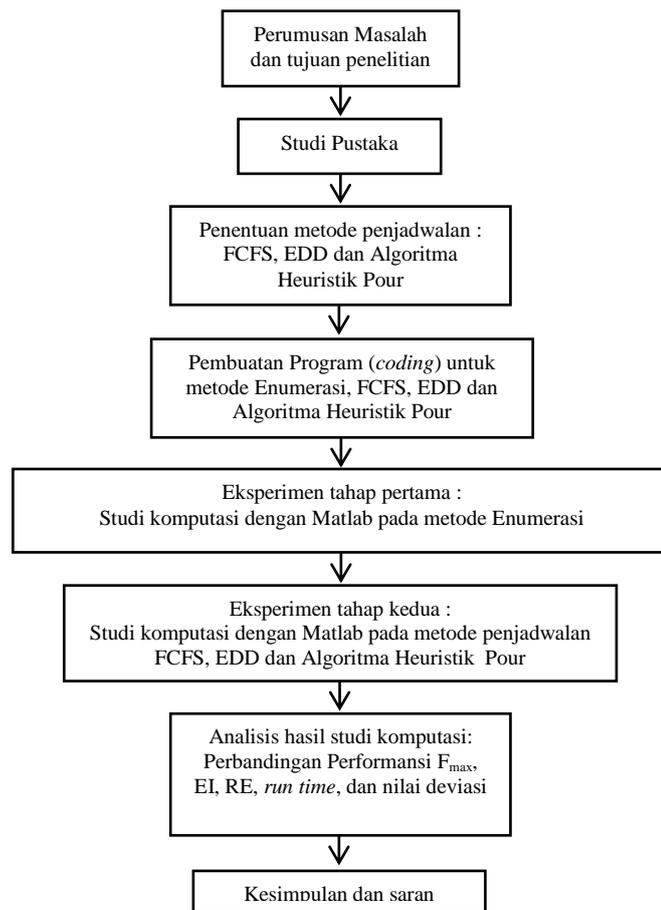
Kemudian data diproses menggunakan metode FCFS, EDD dan Pour untuk meminimalkan *makespan* sebanyak 300 *run*. Dilakukan sebanyak 300 kali *run* karena menurut Arisha et al. (2002), pada tabel A *Steady-State Analysis of the Model* menunjukkan bahwa *number of run* mulai pada kasus 300 *run* mengalami hasil yang stabil.

Penelitian ini dilakukan dalam dua tahap. Tahap pertama adalah melakukan perhitungan semua kombinasi *job* untuk menemukan solusi optimum pada permasalahan *flow shop* untuk meminimalkan *makespan* (Metode Enumerasi). Pada tahap kedua dilakukan eksperimen dengan metode FCFS, EDD dan Pour untuk penjadwalan *flow shop*. Diagram alir penelitian ditunjukkan gambar 1.

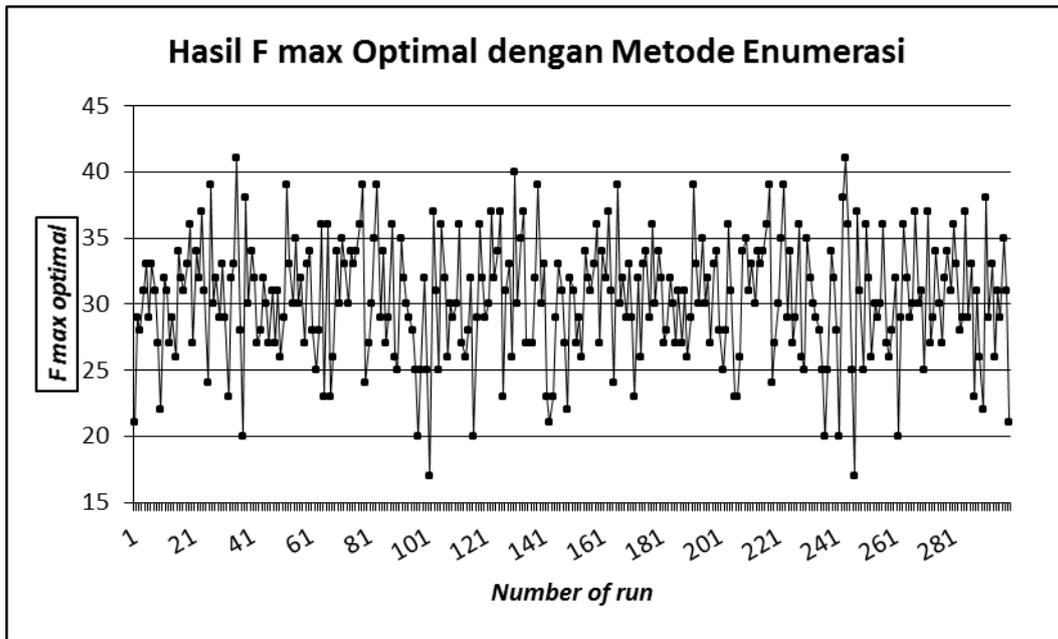
IV. ANALISA HASIL

Data waktu proses dan data *due date* pada penelitian ini merupakan data random yang dibangkitkan dengan *software* Matlab 7.1. Eksperimen dilakukan menggunakan komputer dengan spesifikasi Intel Atom D2500 1.8 Ghz, 2GB DDR3. Data waktu proses memiliki interval $U(1,10)$ dan data *due date* memiliki interval $U(11,20)$. Data random untuk waktu proses dan data *due date* dibangkitkan sampai 300 *run* (Arisha et al, 2002).

Hasil *makespan* optimal dalam 300 *run* dengan metode enumerasi ditunjukkan dalam gambar 2 di bawah ini. Hasil *makespan* optimal kemudian dihitung rata-ratanya. Hasil perhitungan *makespan* rata-rata metode enumerasi sebesar 30,35.



Gambar 1. Diagram Alir Penelitian



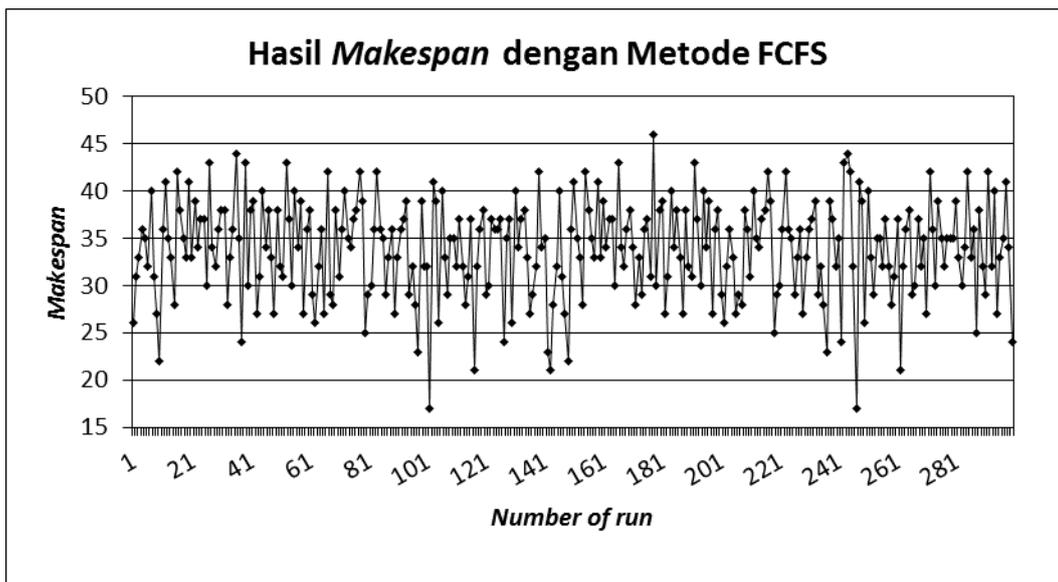
Gambar 2. Hasil Fmax Optimal dengan Metode Enumerasi

Hasil *makespan* dalam 300 *run* dengan metode FCFS ditunjukkan dalam gambar 3 di bawah ini. Hasil *makespan* tersebut kemudian dihitung rata-ratanya. Hasil perhitungan *makespan* metode FCFS sebesar 33,83.

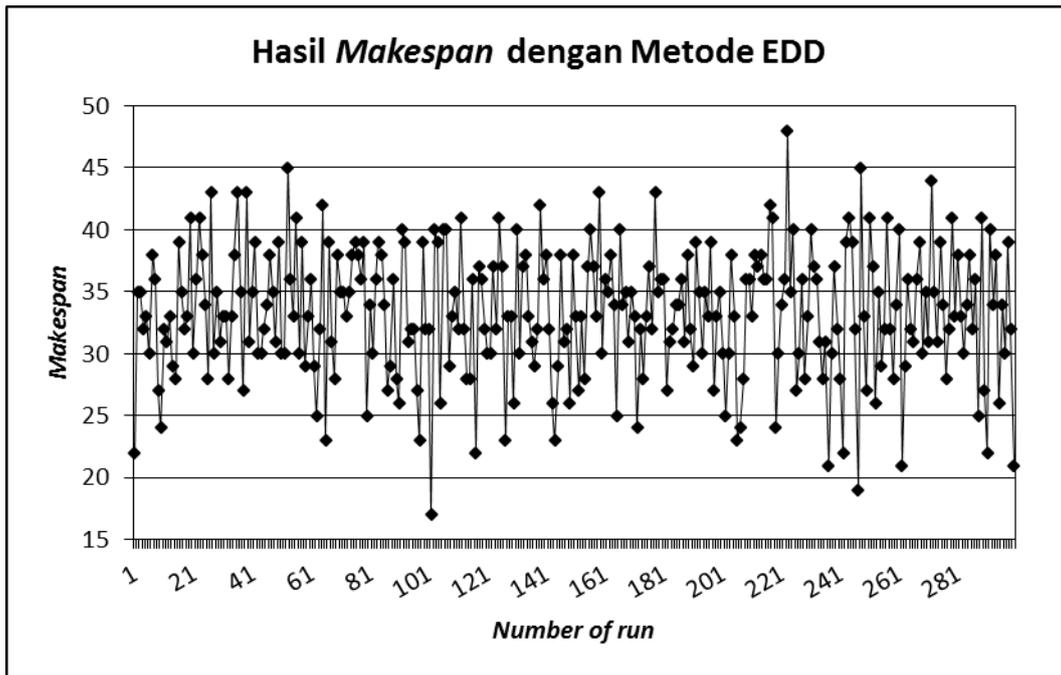
Hasil *makespan* dalam 300 *run* dengan metode EDD ditunjukkan dalam gambar 4 di bawah ini. Hasil *makespan* tersebut kemudian dihitung rata-ratanya. Hasil perhitungan *makespan* metode EDD sebesar 33,29.

Hasil *makespan* dalam 300 *run* dengan metode Pour ditunjukkan dalam gambar 5 di bawah ini. Hasil *makespan* tersebut kemudian dihitung rata-ratanya. Hasil perhitungan *makespan* metode Pour sebesar 30,70.

Hasil *makespan* rata-rata pada metode FCFS, EDD dan Metode Algoritma Heuristik Pour ditampilkan pada tabel 1.



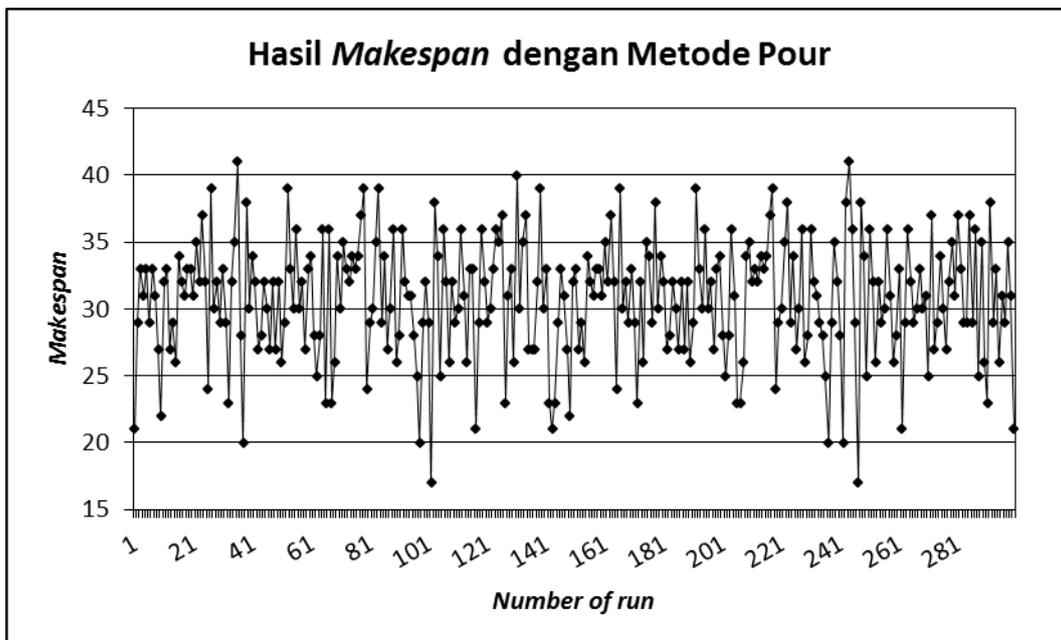
Gambar 3. Hasil *Makespan* dengan Metode FCFS



Gambar 4. Hasil *Makespan* dengan Metode EDD

Tabel 1. Perbandingan *Makespan* Rata-rata metode FCFS, EDD dan Pour

Metode	<i>Makespan</i> Rata-rata (<i>time unit</i>)
Enumerasi	30,35
FCFS	33,83
EDD	33,29
POUR	30,70



Gambar 5. Hasil *Makespan* dengan Metode Pour

Berdasarkan tabel perbandingan *makespan* rata-rata metode FCFS, EDD dan Pour di atas metode FCFS memiliki *makespan* rata-rata yaitu 33,83 *time unit*. Sedangkan metode EDD memiliki *makespan* rata-rata sebesar 33,29 *time unit*. Metode Pour memiliki *makespan* rata-rata sebesar 30,70 *time unit*. Dari hasil tersebut terlihat bahwa metode Pour memiliki *makespan* rata-rata terkecil dari metode FCFS dan EDD yaitu 30,70.

Hasil perhitungan parameter performansi pada *Efficiency Index* (EI) antara metode FCFS dan metode Pour adalah 0,9075. Karena nilai $EI < 1$ maka metode algoritma heuristik Pour memiliki performansi yang lebih baik dari pada metode FCFS. Hasil perhitungan *Efficiency Index* (EI) antara metode EDD dan metode Pour adalah 0,9222. Karena nilai $EI < 1$ maka metode algoritma heuristik Pour memiliki performansi yang lebih baik dari pada metode EDD.

Sedangkan hasil perhitungan *Relative Error* (RE) antara metode FCFS dan metode Pour adalah 10,19 %. Dari hasil perhitungan parameter RE tersebut didapatkan nilai cukup besar atau lebih dari 5% yang menunjukkan bahwa penggunaan metode FCFS akan memiliki perbedaan nilai *makespan* yang cukup besar dibandingkan dengan metode Pour. Hasil perhitungan *Relative Error* (RE) antara metode EDD dan metode Pour adalah 8,44 %. Dari hasil perhitungan parameter RE tersebut juga didapatkan nilai cukup besar atau lebih dari 5% yang menunjukkan bahwa penggunaan metode EDD akan memiliki perbedaan nilai *makespan* yang cukup besar pula dibandingkan dengan metode Pour.

Run time digunakan untuk mengetahui efisiensi sebuah program ketika dijalankan. Tabel 2 menampilkan perbandingan *run time* dari metode FCFS, EDD dan Metode Algoritma Heuristik Pour.

Tabel 2. Perbandingan *Run Time* pada Metode FCFS, EDD dan Pour

Metode	<i>Run time</i> (detik)
FCFS	0.077761
EDD	0.151779
POUR	2.741351

Tabel di atas menampilkan perbandingan nilai *run time* pada masing-masing metode. Metode FCFS memiliki nilai *run time* sebesar 0.077761 detik. Metode EDD memiliki nilai *run time* sebesar 0.151779 detik. Metode Pour memiliki nilai *run time* sebesar 2.741351 detik. Dari hasil tersebut terlihat bahwa *run time* paling kecil adalah metode FCFS sedangkan nilai *run time* paling besar adalah metode Pour. Meskipun metode Pour memiliki nilai *run time* yang paling besar akan tetapi nilai *run time* yang diperoleh tersebut masih dapat ditoleransi.

Nilai deviasi *makespan* rata-rata merupakan selisih antara nilai *makespan* rata-rata metode FCFS, EDD dan Pour dengan nilai *makespan* rata-rata optimal.

Tabel 3. Perbandingan Nilai Deviasi pada metode FCFS, EDD dan Pour

Metode	Nilai deviasi (<i>time unit</i>)
FCFS	3,48
EDD	2,94
POUR	0,35

Tabel di atas menunjukkan hasil perhitungan nilai deviasi *makespan* rata-rata antara metode FCFS, EDD dan Pour. Dari tabel tersebut metode FCFS memiliki nilai deviasi yaitu 3,48 *time unit*. Metode EDD memiliki nilai deviasi yaitu 2,94 *time unit*. Metode Pour memiliki nilai deviasi yaitu 0,35 *time unit*. Dari hasil tersebut metode Algoritma Heuristik Pour memiliki nilai deviasi *makespan* rata-rata terkecil dibandingkan dengan metode FCFS dan metode EDD. Sedangkan nilai deviasi *makespan* rata-rata terbesar adalah metode FCFS. Hal ini berarti metode Algoritma Heuristik Pour memiliki performansi yang baik dalam meminimalkan *makespan* karena memiliki nilai deviasi yang terkecil.

V. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan dapat disimpulkan bahwa :

1. Hasil perhitungan metode FCFS memiliki *makespan* rata-rata sebesar 33,83 *time unit*. Metode EDD memiliki *makespan* rata-rata sebesar 33,29 *time unit*. Metode Algoritma Heuristik Pour memiliki *makespan* rata-rata sebesar 30,70 *time unit*. Sedangkan rata-rata *makespan* optimal berdasarkan metode Enumerasi sebesar 30,35 *time unit*.
2. Nilai *Efficiency Index* (EI) antara metode FCFS dengan metode Pour adalah 0,9075. Nilai *Efficiency Index* (EI) antara metode EDD dengan metode Pour adalah 0,9222. Karena nilai $EI < 1$ maka metode algoritma heuristik Pour memiliki performansi yang lebih baik dari pada metode FCFS dan EDD.
3. Nilai *Relative Error* (RE) antara metode FCFS dan metode Pour adalah 10,19 %. Nilai *Relative Error* (RE) antara metode EDD dan metode Pour adalah 8,44 %. Karena hasil perhitungan parameter RE tersebut didapatkan nilai cukup besar atau lebih dari 5% maka menunjukkan bahwa penggunaan metode FCFS dan EDD akan memiliki perbedaan nilai *makespan* yang cukup besar dibandingkan dengan metode algoritma heuristik Pour.
4. Metode FCFS memiliki nilai *run time* sebesar 0.077761 detik. Metode EDD memiliki nilai *run time* sebesar 0.151779 detik. Metode Pour memiliki nilai *run time* sebesar 2.741351 detik.
5. Hasil perhitungan metode FCFS memiliki nilai deviasi 3,48 *time unit*. Metode EDD memiliki nilai deviasi 2,94 *time unit*. Metode Pour memiliki nilai deviasi 0,35 *time unit*.
6. Dalam penelitian ini metode terbaik untuk penjadwalan *flow shop* n job m mesin dalam meminimalkan *makespan* adalah Algoritma Heuristik Pour. Metode algoritma heuristik Pour lebih baik karena memiliki nilai *makespan* rata-rata terkecil yaitu 30,70 *time unit*, nilai $EI < 1$, nilai RE lebih dari 5%, dan nilai deviasi yang terkecil yaitu 0,35 *time unit*.
7. Meskipun metode Algoritma Heuristik Pour memiliki nilai *run time* yang paling lama akan tetapi nilai *run time* yang diperoleh masih dapat ditoleransi yaitu sebesar 2.741351 detik.

VI. DAFTAR PUSTAKA

- [1] Arisha, A., Young, P., El Baradie, M., 2002. *Flow Shop Scheduling Problem : a Computational Study*. Sixth International Conference on Production Engineering and Design for Development (PEDD6), Cairo Egypt, pp 543-557.
- [2] Baker, K.R. dan Trietsch, D., 2009. *Principles of Sequencing and Scheduling*. Canada : John Willey and Son.
- [3] Bancila, D. dan Buzatu, C., 2014. *Minimizing Makespan In A Flow Shop Scheduling Problem*. Transilvania University of Brasov, Vol. 15, No. 3. pp.43.
- [4] Bedworth, D.D. 1987, *Integrated Production Control Systems, Management, Analysis, Design 2/E*, John Wiley & Sons, New York.
- [5] Ginting, R., 2009. *Perancangan Produk*. Medan : Graha Ilmu.
- [6] Hamman, M.K., 2015. *Penjadwalan Produksi Flow Shop untuk Meminimalkan Makespan dengan Metode Campbell, Dudek and Smith (CDS), Metode Palmer, Metode Dannenbring dan Metode Ignall-Scharge (Studi Kasus di CV. Bonjour Jaya, Klaten)*. S-1. Teknik Industri, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta.
- [7] Heizer, J. dan Render, B., 2010. *Manajemen Operasi*. Jakarta : Salemba Empat.
- [8] Hejazi, S.R., Emami, S., Arkan, A., 2009. A Heuristic Algorithm for Minimizing the Expected in Two-Machine Flow Shop with Fuzzy Processing Time. Department of Industrial & Systems Engineering, Isfahan University of Technology, Vol. 3, No. 2, pp. 114-122.
- [9] Kuncoro, C., 2013. *Penjadwalan Produksi Kertas Menggunakan Algoritma Pour dan Algoritma NEH di PT. Kertas Leces Probolinggo*. S-1. Matematika, Universitas Jember.
- [10] Kusuma, H., 2009. *Perancangan dan Pengendalian Produksi*. Yogyakarta : Penerbit ANDI.

-
- [11] Modrak, V. dan Pandian, R.S., 2010. *Flow Shop Scheduling Algorithm to Minimize Completion Time For n Job m Machine Problem*. Technical Gazette, Vol. 17, pp. 273-278.
- [12] Nasution, A. H. dan Prasetyawan, Y., 2008. *Perancangan dan Pengendalian Produksi*. Yogyakarta : Graha Ilmu.
- [13] Novriansyah, R., 2013. *Peningkatan Perencanaan Penjadwalan Produksi dengan Metode Campbell Dudek and Smith*. S-1. Teknik Industri, Universitas Bina Darma Palembang.
- [14] Pour, H. D., 2001. *A New Heuristic for n-Job m-Machine Flowshop Problem*. Production Planning Control, Vol. 12, No. 7, 648-653.
- [15] Rasjidin, R. dan Hidayat, I., 2006. *Penjadwalan Produksi Mesin Injection Moulding pada PT. Duta Flow Plastic Machinery*. Jurnal Inovasi, Vol. 5, No. 2, pp. 52-57.
- [16] Riska. 2013. *Penerapan Kombinatorial pada Tanda Nomor Kendaraan Bermotor (TNKB) Kota Bandung*. S-1. Teknik Informatika, Institut Teknologi Bandung.
- [17] Soetanto, T.V. dan Palit, H.C., 2004. *Studi Perbandingan Performance Algoritma Heuristik Pour terhadap Mixed Integer Programming dalam Menyelesaikan Penjadwalan Flowshop*. Jurnal Teknik Industri Vol. 6, No. 1, pp. 79-85.
- [18] Stevenson, W. J., 2007. *Operation Management*. New York : McGraw-Hill.
- [19] Sulaksmi, A., Garside, A.K., dan Hadziqah, F. 2014. *Penjadwalan Produksi dengan Algoritma Heuristik Pour (Studi Kasus: Konveksi One Way–Malang)*. Jurnal Teknik Industri, Vol. 15, No. 1, pp. 35–44.