

Penjadwalan Produksi dengan Menggunakan Algoritma Simulated Annealing

by Leo Santoso

Submission date: 10-Oct-2019 06:11PM (UTC+0700)

Submission ID: 1189968966

File name: Paper.pdf (1.99M)

Word count: 2773

Character count: 15308

PENJADWALAN PRODUKSI DENGAN MENGGUNAKAN ALGORITMA *SIMULATED ANNEALING*

Leo Willyanto Santoso^[1], Jonathan Guntara^[2], Iwan Njoto Sandjaja^[3]
Jurusan Teknik Informatika, Fakultas Teknologi Industri
Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya 60236
Telp.: +62-31-2983455, e-mail: ^[1]leow@petra.ac.id, ^[3]iwanns@petra.ac.id

Abstract

X company is a company which produces incense as their main goal. They are using pre-ordering system to do their production process from semi-finished goods to finished goods. By the reason of the same steps of process in all goods production, this process can be categorized as a flowshop. X company still using the FIFO method, so many bottleneck still occurred. In the end, it will affect the increase in makespan. Therefore scheduling application with Simulated Annealing Algorithm is needed to minimize the bottleneck, so the makespan will automatically decrease.

Simulated Annealing Algorithm is one of the heuristic method in search of the optimal solution. Optimal solution was obtained by doing job switching with some unique rule called pairwise exchange. Before doing it, we must look for an initial solution with Shortest Processing Time (SPT), Longest Processing Time (LPT), dan First Job Smallest First (FJSF) methods. Initial solution will also be found by FIFO and Brute Force.

Experimental result shows that Simulated Annealing Scheduling with 15 jobs (average value of X company's job order in 1 week) able to save more than 300 minutes compared to FIFO.

Keyword: *Production scheduling, scheduling, simulated annealing*

1. PENDAHULUAN

Perusahaan X adalah sebuah perusahaan yang bergerak di bidang produksi dupa. Perusahaan ini terletak di Jalan Kyai Tambak Deres no. 96, Surabaya. Sistem produksi dari *raw material* menuju barang setengah jadi dilakukan secara massal, sedangkan dari barang setengah jadi menuju barang jadi dilakukan secara *pre-order*. Proses produksi dari barang setengah jadi menuju barang jadi meliputi beberapa tahap yaitu proses penyerbukan, proses pembuatan motif, proses pengeringan 1, proses pewarnaan dan pemberian

aroma, proses pengeringan 2, proses pemberian *sticker*, dan proses *packing*. Kendala terdapat pada pemenuhan proses *pre-order*. Sistem penjadwalan yang masih memakai metode *First In First Out* (FIFO) belum menunjukkan hasil yang optimal. Hal ini dapat dilihat dari adanya *bottleneck* yang terjadi pada beberapa mesin. Barang setengah jadi dari mesin sebelumnya harus menunggu terlalu lama untuk diproses. Hal ini yang kemudian membawa dampak lamanya waktu pengerjaan total (*makespan*). Karena itu, perusahaan membutuhkan sebuah *tools* untuk membantu menentukan jadwal produksi yang tepat. *Tools* ini berupa aplikasi penjadwalan produksi menggunakan algoritma *Simulated Annealing*. Fokus utama dari *tools* ini adalah meminimalkan *makespan* total yang dengan otomatis juga akan meminimalkan *bottleneck*.

Algoritma *Simulated Annealing* merupakan salah satu metode *heuristic* dalam pencarian solusi optimal [1, 2, 3]. *Simulated Annealing* adalah satu dari algoritma – algoritma terbaik saat ini dalam hampir semua pemecahan masalah optimasi [3]. Algoritma ini memecahkan masalah optimasi dengan mensimulasikan bagaimana logam didinginkan sampai ke struktur kristalnya. Dengan sedikit modifikasi, *Simulated Annealing* akan bekerja sangat baik dalam semua permasalahan optimasi. Penelitian sebelumnya mengatakan bahwa *Simulated Annealing* menghasilkan solusi yang sangat bagus dan mendekati optimal untuk masalah *n-job, m-machine flowshop*. Dibandingkan dengan metode lainnya, algoritma ini mengungguli metode – metode *heuristic* lainnya [4].

Rumusan masalah yang dihadapi dalam penelitian ini adalah bagaimana membuat aplikasi yang bisa menentukan urutan penjadwalan produksi yang terbaik untuk perusahaan X.

Tujuan penelitian ini adalah membuat aplikasi penjadwalan produksi dengan algoritma *Simulated Annealing* sehingga membantu perusahaan X menentukan urutan penjadwalan produksi yang tepat.

2. METODE

Langkah – langkah penelitian ini adalah:

1. Studi Literatur
Mempelajari konsep penjadwalan dengan menggunakan algoritma *Simulated Annealing*.
2. Pengumpulan Data
Melakukan survei perusahaan dan mengumpulkan data dari perusahaan X.
3. Analisis dan desain sistem
Menganalisis cara kerja perusahaan dan dilanjutkan dengan mendesain *Entity Relationship Diagram* (ERD) dan *Data Flow Diagram* (DFD) [5].
4. Implementasi Sistem
Mimplementasikan ERD dan DFD tersebut ke dalam pembuatan sistem.
5. Pengujian Sistem
Melakukan pengujian sistem keseluruhan, apakah sudah cocok dengan yang dibutuhkan oleh perusahaan.

Ruang lingkup dibatasi pada:

1. Pembuatan aplikasi penjadwalan produksi barang. Data yang akan digunakan adalah data kapasitas mesin, data tipe dupa yang diproduksi saat ini, dan contoh data order customer dalam tenggang waktu tertentu.
2. Data yang sudah terkumpul kemudian akan diproses menggunakan algoritma *Simulated Annealing*. Sebagai pembanding, data juga akan diproses menggunakan metode FIFO dan metode *Brute Force*.
3. Hasil berupa perbandingan *makespan* dan waktu proses yang didapat dari ketiga metode penjadwalan tersebut. *Makespan* yang terkecil akan diambil dan dijadikan sebagai penjadwalan terbaik. Sehingga dapat diterapkan oleh perusahaan untuk hasil yang optimal.
4. Pembuatan *software* ini menggunakan program Microsoft Visual Studio 2005 dan *database* Microsoft SQL Server 2005.
5. Diasumsikan bahan setengah jadi yang digunakan telah tersedia sebelum proses produksi berlangsung.
6. Tidak memperhitungkan waktu perpindahan *work in process* (WIP).

Mesin – mesin dan alat bantu produksi telah tersedia, dan semuanya dalam keadaan baik.

3. DISKUSI

Algoritma SA ini diawali dengan pencarian solusi optimal dengan menggunakan metode lain,

dalam hal ini adalah *Shortest Processing Time* (SPT), *Longest Processing Time* (LPT), dan *First Job Smallest First* (FJSF) sebagai solusi optimal awal pada temperatur tinggi. Nilai temperatur tersebut secara perlahan–lahan diturunkan untuk mencari global optimal.

Langkah pertama adalah menentukan suatu nilai temperatur tertentu (dalam hal ini bertemperatur tinggi) agar probabilitas penerimaan tinggi. Perhitungan akan berhenti bila probabilitas penerimaan ini kecil sekali atau sistem telah jenuh. Oleh karena itu maka harus dipilih temperatur yang besar di awal.

$$P = \exp\left(\frac{-\delta}{T}\right) \quad (1)$$

Di mana P adalah probabilitas penerimaan, δ adalah selisih nilai *makespan* baru dengan *makespan* awal dan T adalah temperatur.

Langkah – langkah dari pengerjaan algoritma *Simulated Annealing* ini adalah:

1. Memasukkan nilai temperatur dan faktor pereduksinya (delta temperatur).
2. Melakukan *random* solusi awal dengan menggunakan metode *Shortest Processing Time* (SPT), *Longest Processing Time* (LPT), atau *First Job Smallest First* (FJSF).
3. Menghitung *makespan* awal sebagai solusi optimal sementara.

Di sini diberikan contoh permasalahan. Data contoh permasalahan dapat dilihat pada Tabel 1. kemudian, Table2, Tabel 3 dan Tabel 4 menampilkan penghitungan nilai *Makespan* untuk SPT, LPT dan FJSF.

Tabel 1. Data Waktu Proses Contoh permasalahan

Keterangan	Mesin 1	Mesin 2	Mesin 3	Total
Job 1	10	50	35	95
Job 2	25	15	50	90
Job 3	45	20	15	80
Job 4	15	20	50	85

Tabel 2. Penghitungan Nilai *Makespan* untuk SPT

SPT	Mesin 1		Mesin 2		Mesin 3	
	Start	End	Start	End	Start	End
Job 3	0	45	45	65	65	80
Job 4	45	60	60	85	85	135
Job 2	60	85	85	100	135	185
Job 1	85	95	100	105	185	220

Tabel 3. Penghitungan Nilai *Makespan* untuk LPT

SPT	Mesin 1		Mesin 2		Mesin 3	
	Start	End	Start	End	Start	End
Job 1	0	10	10	60	60	95
Job 2	10	35	60	75	95	145

Job 4	35	50	75	95	145	195
Job 3	50	95	95	115	195	210

Tabel 4. Penghitungan Nilai *Makespan* untuk FJSF

SPT	Mesin 1		Mesin 2		Mesin 3	
	Start	End	Start	End	Start	End
Job 1	0	10	10	60	60	95
Job 4	10	25	60	80	95	145
Job 2	25	50	80	95	145	195
Job 3	50	95	95	115	195	210

Jadi nilai *makespan* untuk aturan SPT adalah sebesar 220 satuan waktu, untuk aturan LPT adalah sebesar 210 satuan waktu, dan untuk aturan FJSF adalah sebesar 210 satuan waktu.

- Mencari inisiasi *neighbourhood* dengan *pairwise exchange*. Misalkan setelah perandoman ternyata keluar angka 1 dan 2, berarti *job* urutan 1 ditukarkan dengan urutan ke 2, sehingga jadwal sekarang menjadi 2 – 1 – 4 – 3 (untuk metode LPT).
- Menghitung *makespan* dari jadwal yang baru. Pada contoh permasalahan di atas, ditemukan jadwal baru setelah *pairwise exchange* yaitu 2 – 1 – 4 – 3. Setelah dihitung, *makespan*-nya adalah sebesar 190 satuan waktu.
- Menghitung nilai δ , jika δ bernilai positif maka ke langkah 7, dan bila negatif maka ke langkah 10.
- Menghitung nilai P (probabilitas penerimaan).
- Mencari bilangan random 0.1 – 0.9.
- Mencari nilai B di mana $B = P \times$ bilangan random, jika $B \leq 0.5$ maka ke langkah 11, jika tidak ke langkah 10.
- Jadwal baru ditentukan sebagai solusi optimal sementara.
- Mencari temperatur baru, jika temperatur telah sesuai dengan yang diinginkan atau sistem telah jenuh maka solusi optimal telah diperoleh, jika tidak kembali ke langkah ke 4.

4. HASIL

Pengujian ini akan berkonsentrasi pada 2 hal penting, yaitu lama proses yang dibutuhkan dan hasil akhir yang dihasilkan oleh suatu algoritma. Jumlah *job* yang akan digunakan untuk pengujian adalah 5 *job*, 6 *job*, 7 *job*, dan 15 *job*.

Pengujian Sistem dengan Menggunakan 5 *job*

Pengujian pertama dilakukan dengan menggunakan 5 *job*.

Penjadwalan pertama menggunakan algoritma FIFO. Lama proses yang dihasilkan adalah 00:00:00:00 yang berarti prosesnya memakan waktu < 1 *centisecond*. Sedangkan *makespan* yang dihasilkan adalah 1805 menit.

Penjadwalan kedua dilakukan dengan algoritma *Simulated Annealing*. Penjadwalan dilakukan sebanyak 5 kali untuk mengetahui apakah jadwal yang dihasilkan sudah yang terbaik. Pada 5 kali percobaan ini, suhu awal dan delta t yang digunakan adalah maksimal, yaitu 1000 C dan 0.95. Lama proses dan *makespan* yang dihasilkan pada percobaan 1 adalah 00:00:11:24 dan 1675 menit.

Lama proses dan *makespan* percobaan 1 – 5 dapat dilihat pada Tabel 5.

Tabel 5. Percobaan 1 – 5 SA 5 *job*

	Lama Proses	Makespan
Percobaan 1	00:00:11:24	1675
Percobaan 2	00:00:08:18	1675
Percobaan 3	00:00:08:24	1675
Percobaan 4	00:00:08:48	1675
Percobaan 5	00:00:09:12	1675

Penjadwalan ketiga dilakukan dengan algoritma *Brute Force* yang akan mencari semua kemungkinan urutan jadwal yang berjumlah 120. Lama proses yang dibutuhkan adalah 00:00:11:33. Sedangkan *makespan* yang dihasilkan adalah 1675 menit.

Dari hasil pengujian 3 algoritma menggunakan 5 *job*, dapat diambil beberapa kesimpulan sebagai berikut:

- Lama proses SA lebih lama dibandingkan FIFO, tetapi *makespan* yang dihasilkan lebih singkat.
- Lama proses SA lebih singkat dibandingkan *Brute Force*, tetapi *makespan* yang dihasilkan 100% sama dengan yang dihasilkan *Brute Force*.

Pengujian Sistem dengan Menggunakan 6 *job*

Pengujian kedua dilakukan dengan menggunakan 6 *job*.

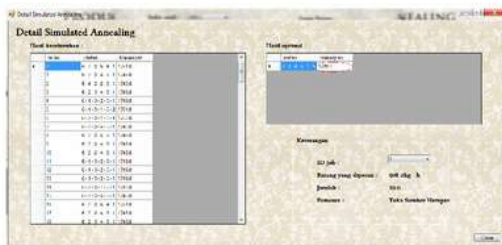
Penjadwalan pertama menggunakan algoritma FIFO. Lama proses yang dihasilkan adalah 00:00:00:00 yang berarti prosesnya memakan waktu < 1 *centisecond*. Sedangkan *makespan* yang dihasilkan adalah 1855 menit.

Penjadwalan kedua dilakukan dengan algoritma *Simulated Annealing*. Penjadwalan dilakukan sebanyak 5 kali untuk mengetahui apakah jadwal yang dihasilkan sudah yang terbaik. Pada 5 kali percobaan ini, suhu awal dan delta t yang digunakan adalah maksimal, yaitu 1000 C

dan 0.95. Lama proses dan *makespan* yang dihasilkan pada percobaan 1 dapat dilihat pada Gambar 1. dan Gambar 2.



Gambar 1. Lama Proses SA 6 job – 1



Gambar 2. Makespan SA 6 job – 1

Sedangkan lama proses dan *makespan* percobaan 1 – 5 dapat dilihat pada Tabel 6.

Tabel 6. Percobaan 1 – 5 SA 6 job

	Lama Proses	Makespan
Percobaan 1	00:00:09:39	1729
Percobaan 2	00:00:08:72	1729
Percobaan 3	00:00:11:21	1729
Percobaan 4	00:00:10:48	1745
Percobaan 5	00:00:10:27	1729

Penjadwalan ketiga dilakukan dengan algoritma *Brute Force* yang akan mencari semua kemungkinan urutan jadwal yang berjumlah 720. Lama proses yang dibutuhkan adalah 00:02:19:42. Sedangkan *makespan* yang dihasilkan adalah 1729 menit.

Dari hasil pengujian 3 algoritma menggunakan 6 job, dapat diambil beberapa kesimpulan sebagai berikut:

- Lama proses SA lebih lama dibandingkan FIFO, tetapi *makespan* yang dihasilkan lebih singkat.

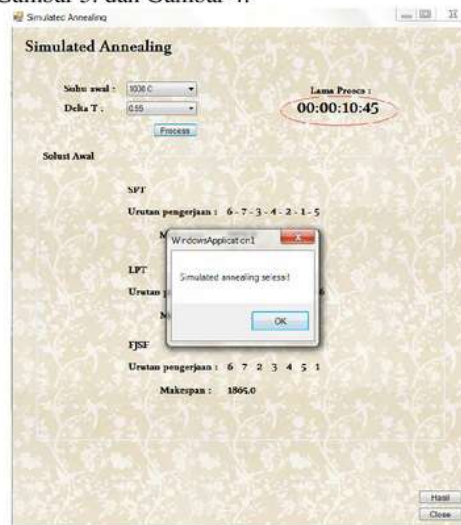
- Lama proses SA sekitar 12 kali lebih singkat dibandingkan *Brute Force*, tetapi *makespan* yang dihasilkan 80% sama dengan yang dihasilkan *Brute Force*.

Pengujian Sistem dengan Menggunakan 7 job

Pengujian ketiga dilakukan dengan menggunakan 7 job.

Penjadwalan pertama menggunakan algoritma FIFO. Lama proses yang dihasilkan adalah 00:00:00:00 yang berarti prosesnya memakan waktu < 1 centisecond. Sedangkan *makespan* yang dihasilkan adalah 1907 menit.

Penjadwalan kedua dilakukan dengan algoritma *Simulated Annealing*. Penjadwalan dilakukan sebanyak 5 kali untuk mengetahui apakah jadwal yang dihasilkan sudah terbaik. Pada 5 kali percobaan ini, suhu awal dan delta t yang digunakan adalah maksimal, yaitu 1000 C dan 0.95. Lama proses dan *makespan* yang dihasilkan pada percobaan 1 dapat dilihat pada Gambar 3. dan Gambar 4.



Gambar 3. Lama Proses SA 7 job – 1



Gambar 4. Makespan SA 7 job – 1

Sedangkan lama proses dan *makespan* percobaan 1 – 5 dapat dilihat pada Tabel 7.

Tabel 7. Percobaan 1 – 5 SA 7 job

	Lama Proses	Makespan
Percobaan 1	00:00:10:45	1799
Percobaan 2	00:00:10:54	1799
Percobaan 3	00:00:10:45	1799
Percobaan 4	00:00:10:39	1799
Percobaan 5	00:00:10:57	1815

Penjadwalan ketiga dilakukan dengan algoritma *Brute Force* yang akan mencari semua kemungkinan urutan jadwal yang berjumlah 5040. Lama proses yang dibutuhkan adalah 00:25:09:87. Sedangkan *makespan* yang dihasilkan adalah 1799 menit.

Dari hasil pengujian 3 algoritma menggunakan 7 job, dapat diambil beberapa kesimpulan sebagai berikut:

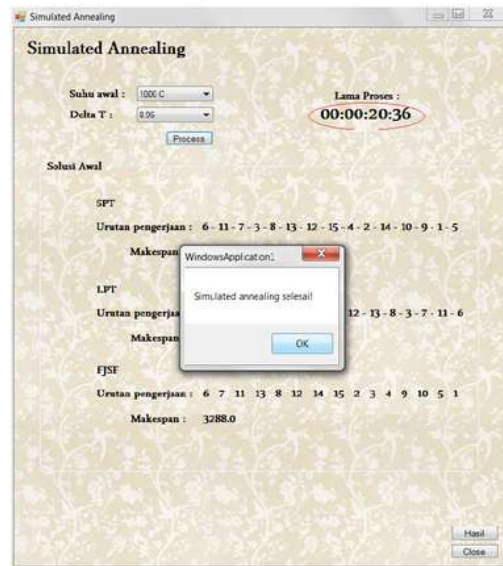
- Lama proses SA lebih lama dibandingkan FIFO, tetapi *makespan* yang dihasilkan lebih singkat.
- Lama proses SA sekitar 150 kali lebih singkat dibandingkan *Brute Force*, tetapi *makespan* yang dihasilkan 80% sama dengan yang dihasilkan *Brute Force*.

Pengujian Sistem dengan menggunakan 15 job

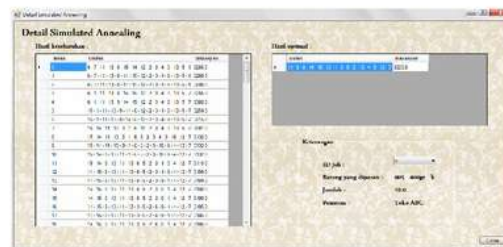
Pengujian keempat dilakukan dengan menggunakan 15 job (rata – rata jumlah job yang dikerjakan oleh perusahaan X dalam 1 minggu).

Penjadwalan pertama menggunakan algoritma FIFO. Lama proses yang dihasilkan adalah 00:00:00:00 yang berarti prosesnya memakan waktu < 1 centisecond. Sedangkan *makespan* yang dihasilkan adalah 3385 menit.

Penjadwalan kedua dilakukan dengan algoritma *Simulated Annealing*. Karena jumlah job yang banyak, maka penjadwalan dilakukan sebanyak 10 kali untuk mengetahui apakah jadwal yang dihasilkan sudah yang terbaik. Pada 10 kali percobaan ini, suhu awal dan delta t yang digunakan adalah maksimal, yaitu 1000 C dan 0.95. Lama proses dan *makespan* yang dihasilkan pada percobaan 1 dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Lama Proses SA 15 job – 1



Gambar 6. Makespan SA 15 job – 1

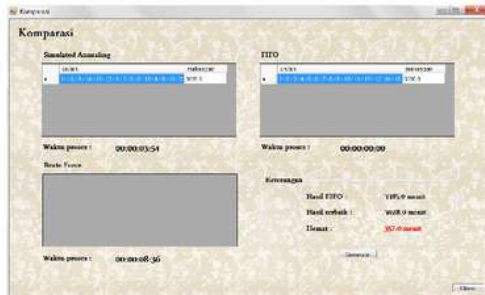
Sedangkan lama proses dan *makespan* percobaan 1 – 10 dapat dilihat pada Tabel 8.

Tabel 8. Percobaan 1 – 10 SA 15 job

	Lama Proses	Makespan
Percobaan 1	00:00:20:36	3028
Percobaan 2	00:00:20:45	3028
Percobaan 3	00:00:20:81	3029
Percobaan 4	00:00:19:39	3028
Percobaan 5	00:00:20:09	3029
Percobaan 6	00:00:19:69	3051
Percobaan 7	00:00:20:45	3038
Percobaan 8	00:00:19:51	3051
Percobaan 9	00:00:19:21	3028
Percobaan 10	00:00:20:69	3028

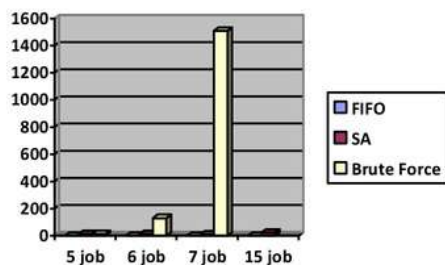
Berhubung pada pengujian sebelumnya (*Brute Force 7 job*), yang dengan tujuan mencari 5040 kemungkinan urutan job saja, memakan waktu yang sangat lama (00:25:09:87). Maka tidak memungkinkan bila dilakukan pengujian *Brute Force 15 job* yang memiliki 1.307.674.368.000 kemungkinan.

Dari hasil pengujian 2 algoritma menggunakan 15 *job*, dapat diambil kesimpulan bahwa lama proses SA lebih lama dibandingkan FIFO, tetapi *makespan* yang dihasilkan jauh lebih singkat. Perbandingan antara FIFO dan SA secara jelas dapat dilihat pada *form* komparasi pada Gambar 7.



Gambar 7. Komparasi FIFO dan SA 15 *job*

Pada Gambar 8. dijelaskan melalui grafik hasil perbandingan waktu yang dibutuhkan 3 algoritma tersebut untuk menghitung beberapa proses dengan beberapa *job*.



Gambar 8. Grafik Perbandingan 3 algoritma

Hasil dari perancangan dan pembuatan aplikasi penjadwalan produksi pada perusahaan X dengan metode *Simulated Annealing* ini dapat diambil beberapa kesimpulan sebagai berikut:

- Algoritma *Simulated Annealing* dapat diterapkan dengan baik dalam penjadwalan

produksi yang digolongkan ke dalam *flowshop* ini. Tidak hanya dapat diterapkan algoritma ini bahkan sangat efektif, karena mampu mengatasi penjadwalan dengan jumlah *job* banyak (15 *job*) dengan waktu proses yang relative cepat dan hasil penjadwalan yang baik, yang tidak dapat diatasi oleh algoritma *Brute Force*.

- Dengan aplikasi penjadwalan ini, perusahaan X dapat menghemat lebih dari 300 menit untuk tiap 15 *job* (rata – rata jumlah *job* yang dikerjakan oleh perusahaan X dalam 1 minggu).
- Aplikasi dapat memberikan informasi mengenai *job* yang harus dikerjakan setiap harinya.

5. DAFTAR PUSTAKA

- [1] T. Morton, and D.W. Pentico, *Heuristic Scheduling Systems: With Application to Production Systems and Project Management*, John Wiley & Sons, New York, 1993.
- [2] F.A. Ogbu and D.K. Smith, “Simulated Annealing for the Permutation Flowshop Problem”, *Omega The International Journal of Management Science*, 19(1), pp. 64-67, 1990.
- [3] N. Schimdt, (2004), *Simulated Annealing*, www.denison.edu/academics/departments/mathcs.
- [4] M.L. Pinedo, *Scheduling: Theory, Algorithm and Systems*, 3rd Ed. Prentice Hall, New Jersey, 2008.
- [5] J.L. Whitten, L.D. Bentley, K.C. Dittman, *Systems Analysis and Design Methods*, McGraw-Hill Education, Indianapolis, 2004.

Penjadwalan Produksi dengan Menggunakan Algoritma Simulated Annealing

ORIGINALITY REPORT

7 %

SIMILARITY INDEX

6 %

INTERNET SOURCES

3 %

PUBLICATIONS

1 %

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

1%

★ [ebookbrowse.in](#)

Internet Source

Exclude quotes On

Exclude bibliography On

Exclude matches < 5 words