

# Perbandingan Algoritma Tabu Search dan Algoritma .. Genetika dalam Penjadwalan Mesin untuk Minimasi Biaya Earliness dan Tardiness

*by Dewi Dian Retno Sari*

---

**Submission date:** 20-Apr-2023 11:02AM (UTC+0700)

**Submission ID:** 2070006006

**File name:** 29p-Pebandingan\_algoritma\_tabu\_search.pdf (5.24M)

**Word count:** 3997

**Character count:** 23410

# Perbandingan Algoritma *Tabu Search* dan Algoritma Genetika dalam Penjadwalan Mesin untuk Minimasi Biaya *Earliness* dan *Tardiness*

Dian Retno Sari Dewi, Anastasia Lidya Maukar, Vina  
E-mail : dianretnosd@yahoo.com

**Abstrak**—Penjadwalan merupakan permasalahan sequencing terhadap sejumlah operasi dan mengalokasikannya tanpa melanggar batasan yang dimiliki. Penjadwalan dalam masalah *job shop* seringkali mengalami *earliness* atau *tardiness* sehingga menimbulkan biaya. Penjadwalan dengan metode optimasi membutuhkan biaya besar ditambah dengan waktu yang panjang. Oleh sebab itu dikembangkan metode heuristik yang menghasilkan solusi mendekati optimal dengan waktu yang singkat. Algoritma *Tabu Search* merupakan metode pencarian secara sistematis dan memiliki keunggulan untuk menghindari solusi optimum lokal. Sedangkan algoritma Genetik adalah teknik pencarian stokastik berdasarkan pada seleksi dan genetika alami. Penelitian ini bertujuan untuk membandingkan kedua algoritma ini dalam kasus *job shop* dengan data hipotetik dengan fungsi objektif minimasi biaya *earliness* dan *tardiness*. Dari hasil uji coba diketahui bahwa algoritma *Tabu Search* menghasilkan biaya lebih rendah dibanding algoritma Genetik.

**Kata kunci**—Penjadwalan; algoritma *Tabu Search*; algoritma Genetik

## I. PENDAHULUAN

Penjadwalan adalah alokasi sumber daya dalam suatu rentang waktu untuk menghasilkan sebuah kumpulan pekerjaan. (Baker, 1974). Secara umum, *scheduling* merupakan suatu permasalahan dalam hal melakukan sequencing terhadap sejumlah operasi dan mengalokasikannya dalam slot waktu tertentu tanpa melanggar batasan teknis dan kapasitas yang dimiliki.

Dalam penjadwalan berbasis biaya, dikenal keterlambatan positif (*tardiness*) dan keterlambatan negatif (*earliness*). Kedua macam keterlambatan ini akan menimbulkan biaya, pada *tardiness* karena tidak dapat memenuhi tenggat waktu yang dijanjikan pada konsumen sehingga ada biaya keterlambatan/penalti dan pada *earliness* karena akan menimbulkan biaya simpan di gudang.

Metode optimasi akan menghasilkan solusi yang optimal, tetapi metode optimasi akan membutuhkan waktu komputasi yang sangat lama dalam proses pencarian solusinya. Metode penjadwalan heuristik dapat menghasilkan solusi yang mendekati optimal dengan waktu komputasi yang singkat. Metode heuristik diantaranya *ant colony*, *simulated annealing*, *tabu search* dan genetika. *Ant colony* merupakan teknik probabilistik untuk menyelesaikan masalah dengan meniru

perilaku semut dalam menemukan jalur koloninya menuju makanan. *Simulated annealing* adalah algoritma pencarian solusi berbasis probabilitas dan mekanika statistik untuk masalah-masalah yang ruang solusinya terlalu besar sehingga hampir tidak mungkin ditemukan solusi eksak terhadap masalah tersebut.

Algoritma genetik adalah teknik pencarian stokastik berdasarkan seleksi alam dan genetika alami. Dimulai dengan sekumpulan solusi acak awal yang disebut sebagai populasi. Tiap individu dalam populasi disebut sebagai kromosom, yang merepresentasikan sebuah solusi untuk permasalahan yang dihadapi. (Mitsuo Gen, 1994).

Algoritma *tabu* merupakan metode yang melakukan pencarian solusi secara sistematis. Kelebihan lain dari algoritma ini adalah kemampuannya untuk menghindari solusi optimum lokal dan *cycling*.

S. Webster dan A. Gupta (1998), melakukan penelitian tentang algoritma Genetik pada kasus penjadwalan dengan mesin tunggal dengan kriteria minimasi biaya penalti *earliness* dan *tardiness*. Dalam penelitian itu dibuktikan bahwa algoritma Genetik dapat menghasilkan solusi yang mendekati optimal dengan cara yang mudah dan waktu komputasi yang singkat.

Guohua Wan dan Benjamin Yen (2002) telah melakukan penelitian tentang algoritma *Tabu Search* pada penjadwalan *job shop* mesin tunggal dengan fungsi objektifnya meminimasi total bobot *earliness* dan *tardiness*. Dalam penelitian tersebut dibuktikan bahwa algoritma *Tabu Search* dapat menyelesaikan masalah penjadwalan tersebut dengan sangat baik khususnya pada ukuran *job* yang cukup besar.

Seperti yang telah dinyatakan dalam penelitian Webster (1998) dan Guohua Wan (2002) bahwa algoritma Genetik dan *Tabu Search* merupakan algoritma yang sangat baik dan dapat menghasilkan solusi mendekati optimal dalam kasus *job shop* dengan kriteria minimasi biaya *earliness* dan *tardiness*. Maka dalam penelitian ini akan dilakukan simulasi dengan kedua algoritma tersebut yaitu algoritma Genetik dan *Tabu Search* dengan mesin paralel dan fungsi objektifnya meminimasi biaya *earliness* dan *tardiness*

ditinjau dari waktu penyelesaian tiap *job* untuk membandingkan performansi dari keduanya.

## II. TINJAUAN PUSTAKA

Penjadwalan adalah penentuan waktu mulai dan berakhirnya suatu pekerjaan serta penugasan setiap pekerjaan pada masing-masing stasiun kerja atau mesin yang dibutuhkan atau penentuan waktu penggunaan sumber daya (tenaga kerja, mesin, peralatan) secara spesifik untuk memproduksi barang atau untuk menghasilkan jasa.

Tujuan penjadwalan secara umum :

- Mengurangi waktu menganggur dari sumber daya.
- Mengurangi tingkat persediaan barang setengah jadi.
- Mengurangi keterlambatan (*tardiness*).

### A. Penjadwalan pada Job Shop

Dalam penjadwalan *job shop*, untuk *input* berupa *routing* dan juga waktu proses biasa dituliskan dalam bentuk matriks. Penulisan dalam matriks biasanya menggunakan suatu notasi tertentu, yaitu notasi  $i,j,k$ . Notasi  $i, j, k$  ini disebut sebagai *triplet*. Notasi  $i$  menunjukkan nomor suatu pekerjaan, notasi  $j$  menunjukkan urutan operasi pada suatu *job*, dan notasi  $k$  menunjukkan jenis mesin yang digunakan. Matriks *routing* menyatakan urutan pengerjaan *job*  $i$  untuk setiap operasi  $j$  pada mesin  $k$ . Contoh matriks *routing* dapat dilihat pada Tabel 1.

Tabel 1. Contoh Matriks *Routing*

Operation \ Job	1	2	3
1	$m_{11}$	$m_{12}$	$m_{13}$
2	$m_{21}$	$m_{22}$	$m_{23}$
3	$m_{31}$	$m_{32}$	$m_{33}$

Keterangan :  $m_{11}$  menunjukkan mesin yang digunakan untuk mengerjakan pekerjaan 1 operasi 1, dst.

Matriks waktu proses menyatakan waktu yang diperlukan oleh suatu *job*  $i$  untuk setiap operasi  $j$  diproses di mesin  $k$ . Adapun contoh matriks waktu proses dapat dilihat pada Tabel 2.

Tabel 2. Contoh Matriks Waktu Proses

Operation \ Job	1	2	3
1	$t_{11}$	$t_{12}$	$t_{13}$
2	$t_{21}$	$t_{22}$	$t_{23}$
3	$t_{31}$	$t_{32}$	$t_{33}$

Keterangan :  $t_{11}$  menunjukkan waktu yang diperlukan untuk mengerjakan pekerjaan 1 operasi 1, dst.

Penjadwalan pada *job shop* yang melibatkan  $n$  *job* dan  $m$  mesin akan menghasilkan sejumlah  $(n!)^m$  kombinasi alternatif jadwal. Dengan semakin banyaknya jumlah *job* dan jumlah

jenis mesin yang terlibat akan mengakibatkan semakin kompleksnya penjadwalan yang dilakukan. Kekompleksitasan penjadwalan pada *job shop* ini juga akan meningkat dengan terjadinya kondisi-kondisi dinamis dalam sistem produksi.

### B. Algoritma Tabu Search

*Tabu Search* merupakan suatu metode heuristik yang dimunculkan pertama kali oleh Fred Glover (Reeves). Algoritma ini dirancang untuk memperoleh satu solusi yang *feasible* (layak) dengan menggunakan serangkaian gerakan pencarian (*move*) dari satu solusi lokal ke solusi lokal yang lain. Untuk menghindari terjebaknya pencarian ke dalam suatu solusi optimal lokal tertentu, maka beberapa gerakan disebut tabu (terlarang). Kondisi tabu ini dapat diabaikan bila ada suatu kriteria (*aspiration criteria*) yang memungkinkannya.

Karakteristik dari algoritma ini adalah :

1. Menggunakan *memory* (disebut *adaptive memory*) dalam bentuk *tabu list*.
2. Menggunakan suatu metode yang sistematis (disebut *neighborhood search*) dalam menentukan pergerakan dari satu solusi ke solusi berikutnya.

Setelah diperoleh sebuah solusi, pencarian tidak berhenti sampai di sana. Pencarian akan berhenti setelah mencapai batas maksimum iterasi atau setelah memenuhi suatu kondisi penghentian (*stopping criteria*).

Algoritma *Tabu Search* secara umum (tanpa menggunakan fungsi diversifikasi) dapat dilihat sbd. (Glover) :

1. Tentukan suatu solusi awal yang *feasible*, yaitu  $x^*$ . Hitung nilai fungsi obyektifnya ( $z^*$ ).
2. Solusi terbaik saat ini adalah  $x^*$  dan fungsi obyektif terbaik saat ini adalah  $z^*$ .
3. Lakukan iterasi dengan cara :
  - a. Lakukan *move* dengan mekanisme *neighborhood search* dengan menggunakan operator *neighborhood search* tertentu. Kemudian periksa status *move* apakah *move* tersebut *inadmissible* (berada dalam status *tabu*) atau *admissible* (berada dalam status *non-tabu*).
  - b. Lakukan perubahan (*update*) pada *tabu list*, yaitu dengan memasukkan *move* yang telah dilakukan untuk dinyatakan sebagai *tabu* dan mengeluarkan *move* yang telah bebas dari status *tabu*.
  - c. Pilih *move admissible* terbaik dari *neighborhood* yang telah terbentuk.
  - d. Tentukan solusi terbaik saat ini menjadi  $x'$  dan fungsi obyektif saat ini menjadi  $z(x')$ .
  - e. Iterasi kemudian kembali ke langkah 3a.
  - f. Iterasi akan dihentikan setelah kondisi penghentian (*stopping criteria*) dicapai.

### 7 C. Algoritma Genetika

Algoritma Genetika merupakan suatu metode heuristik yang pertama kali dikembangkan oleh John Holland. Konsep dasar Algoritma Genetika ini diambil dari proses evolusi secara alami (Morton dan Pentico). Dari populasi yang ada saat ini, untuk setiap generasinya dipilih solusi yang baik (individu dengan karakteristik terpilih, kemudian disebut sebagai *parent*) untuk menghasilkan solusi baru (*offspring*), yaitu dengan mengambil karakteristik terbaik dari masing-masing *parent*-nya dan menggabungkan (melakukan *crossover*) karakteristik tersisa lainnya dan atau dengan melakukan mutasi dari karakteristik salah satu *parent*-nya. *Offspring* (solusi) dengan karakteristik terburuk akan punah dan digantikan oleh *offspring* (solusi) yang lain.

Karakteristik dari Algoritma Genetika adalah (Goldberg):

1. Melakukan pencarian solusi dari beberapa titik pada populasi, bukan hanya pada satu titik saja.
2. Pencarian solusi menggunakan aturan transisi yang probabilistik, bukan deterministik.
3. Bekerja dengan kode-kode set parameter, bukan dengan parameter tersebut.

Algoritma genetik adalah teknik pencarian stokastik berdasarkan mekanisme seleksi alami dan genetika alami. Algoritma genetik adalah algoritma iteratif yang menjaga sekumpulan solusi pada tiap iterasi. Pada awalnya, kumpulan solusi dihasilkan secara acak, dan pada tiap iterasi sebuah kumpulan solusi yang baru dibentuk oleh operator genetik yang meniru prinsip evolusi dan hereditas. Tiap solusi dievaluasi dengan sebuah fungsi obyektif, dan proses ini diulang hingga beberapa bentuk konvergensi dicapai. Kumpulan solusi yang baru dapat terdiri dari solusi-solusi lama yang terpilih karena nilai fungsi obyektifnya yang tinggi, dan juga solusi-solusi baru yang terbentuk dengan mengombinasikan solusi-solusi lain.

Algoritma genetik berbeda dengan teknik pencarian konvensional, mulai dengan sekumpulan solusi acak awal yang disebut sebagai populasi. Tiap individu dalam populasi disebut sebagai kromosom, yang merepresentasikan sebuah solusi untuk permasalahan yang dihadapi. Kromosom tersebut berevolusi melalui beberapa iterasi berurutan, yang disebut sebagai generasi. Selama tiap generasi, kromosom dievaluasi, menggunakan beberapa pengukuran *fitness*. Untuk menghasilkan generasi berikutnya, kromosom-kromosom baru, yang disebut sebagai *offspring*, dibentuk dengan cara sebagai berikut:

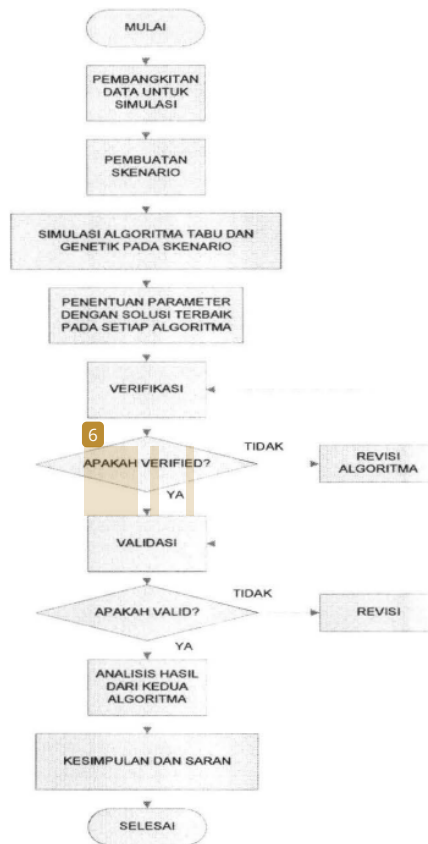
1. Menggabungkan dua kromosom dari generasi sekarang dengan menggunakan operator persilangan.
2. Memodifikasi sebuah kromosom menggunakan sebuah operator mutasi.

Kerangka kerja sebuah algoritma genetik dapat diuraikan sebagai berikut:

1. Inisialisasi. Menghasilkan sebuah populasi string secara acak.
2. Mengevaluasi string berdasarkan fungsi *fitness*.
3. Menggunakan sekumpulan operator genetik untuk menghasilkan populasi string yang baru.

4. Mengulangi langkah ke-2 dan ke-3 hingga sebuah jumlah iterasi yang ditetapkan.

### III. METODOLOGI PENELITIAN



Gambar 1. Diagram Alir Tahapan Penelitian

Metode penelitian yang digunakan dalam penelitian ini meliputi:

1. Pembangkitan data untuk simulasi  
Untuk proses simulasi, data yang dibutuhkan adalah data waktu proses dan data *routing*, supaya waktu proses memiliki sebaran yang merata maka data waktu proses yang digunakan berdistribusi *uniform* dengan *range* satuan waktu antara 1 sampai dengan 10. Sedangkan untuk menentukan suatu operasi dikerjakan pada mesin tertentu dilakukan secara *random*.
2. Pembuatan skenario  
Skenario yang dikembangkan berisi sejumlah variabel tetap dan variabel yang diubah-ubah. Variabel-variabel tersebut adalah:
  - Variabel tetap :

- o Jumlah mesin yaitu sebanyak 2 jenis mesin dengan jumlah unit paralel masing-masing 2 buah mesin.
- o Denda yang dikenakan untuk *earliness* dan *tardiness*, yaitu Rp 50,- untuk denda *earliness* dan Rp 100,- untuk denda *tardiness* per satuan waktu.
- Variabel yang diubah :
  - o Jumlah *job* yaitu jumlah *job* 2 sampai dengan *job* 9.
  - o Jumlah operasi yaitu *job* dengan jumlah operasi 3, 4, 5.

Selain itu, pada tiap algoritma akan dilakukan uji parameter. Pada algoritma Genetik parameter yang diujikan adalah *crossover rate* dan *mutation rate*. *Crossover rate* yang akan diujikan yaitu 0,6; 0,7; 0,8 dan 0,9. *Mutation rate* yang akan diujikan yaitu 0,05; 0,1; 0,15; 0,2; 0,25 dan 0,3. Sedangkan pada algoritma *Tabu Search* parameter yang diujikan adalah ukuran *tabu list*. *Tabu list* yang akan diujikan yaitu *tabu list* dengan ukuran 2 sampai dengan 11. Bentuk umum dari skenario yang dikembangkan adalah sebagai berikut :

Tabel 3.1 Bentuk umum skenario

Algoritma	Parameter	Job 1			Job 2			Job 3			
		Op 1	Op 2	Op 3	Op 1	Op 2	Op 3	Op 1	Op 2	Op 3	
Genetik	Crossover rate ?	Mutation rate 1	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 2	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 3	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
	Crossover rate ?	Mutation rate 1	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 2	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 3	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
	Crossover rate ?	Mutation rate 1	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 2	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
		Mutation rate 3	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$
Tabu	tabu list 1	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	
	tabu list 2	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	
	tabu list 3	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	$B_{ij}$	

Keterangan :

- B = Rata-rata hasil simulasi \*)
- i = Job ke -i
- j = Operasi ke -j
- k = Crossover rate ke -k
- l = Mutation rate ke -l
- m = Tabu list ke -m

\*) Hasil simulasi berupa total biaya *earliness* dan *tardiness* pada job ke-i dan operasi ke-j

3. Simulasi algoritma *Tabu Search* dan Genetik pada setiap skenario  
 Simulasi algoritma dilakukan dengan bantuan program *Delphi 7*. Setiap *job* dan operasi direplikasi sebanyak 10 kali. Pada setiap sub-skenario dilakukan percobaan pada parameter guna mendapatkan parameter terbaik dari setiap algoritma untuk mendapat hasil yang terbaik pula. Pada algoritma *Tabu Search* parameter yang digunakan adalah parameter *tabu list*, sedangkan pada algoritma Genetik parameter yang digunakan adalah *crossover rate* dan *mutation rate*.
4. Penentuan parameter dengan solusi terbaik pada setiap algoritma.

Setelah melakukan replikasi dan percobaan parameter maka diambil hasil terbaik dari setiap sub-skenario tersebut. Hasil terbaik dari setiap algoritma ini yang akan digunakan untuk perbandingan algoritma dan proses analisis.

5. Verifikasi  
 Verifikasi dilakukan dengan mengecek apakah penerapan algoritma telah benar dan juga dilakukan dengan mengecek apakah semua pembatas tidak ada yang dilanggar, yaitu
  - Pembatas urutan operasi
  - Pembatas mesin (tidak boleh mengerjakan 2 *job* pada saat bersamaan)
  - Semua operasi telah dijadwalkan.

Jika dari hasil *output* semua pembatas tidak ada yang dilanggar maka sudah dapat dikatakan *verified*. Jika dari hasil *output* masih ada pembatas yang dilanggar maka harus dilakukan revisi pada algoritma, yaitu dengan mengecek apakah langkah-langkah dalam algoritma sudah benar.

6. Validasi  
 Validasi dilakukan dengan membandingkan hasil dari kedua algoritma dengan hasil optimasi yang dirumuskan dalam program LINGO. Perbandingan ini dilakukan untuk mengetahui seberapa jauh perbedaan hasil antara metode heuristik yang diterapkan dengan hasil optimasi. Jika keduanya tidak memiliki perbedaan yang signifikan maka sudah dapat dikatakan *valid*. Jika terdapat perbedaan yang signifikan maka harus dilakukan revisi yaitu dengan mengecek apakah inputan pada algoritma dan optimasi merupakan inputan yang sama dan benar.
7. Analisis hasil dari kedua algoritma  
 Pada bagian analisis data yang digunakan adalah hasil terbaik dari percobaan parameter pada masing-masing skenario. Analisis yang dilakukan meliputi analisis parameter yaitu untuk menentukan parameter yang menghasilkan biaya terendah dari masing-masing algoritma, analisis *job* dan operasi yaitu untuk membandingkan langsung hasil terbaik dari masing-masing algoritma dan menentukan algoritma mana yang menghasilkan biaya terendah, dan analisa algoritma yaitu untuk menganalisis karakter, kekurangan dan kelebihan dari masing-masing algoritma.
8. Kesimpulan dan saran  
 Membuat kesimpulan dari hasil analisis dan menyertakan saran untuk penelitian lebih lanjut.

A. Pengumpulan dan Pengolahan Data

Bab ini berisi simulasi untuk kedua algoritma dan rekap hasil dari simulasi tersebut. Skenario yang telah dikembangkan disimulasikan pada masing-masing algoritma. Simulasi dilakukan dengan bantuan program *Delphi 7*. Simulasi Program Aplikasi

Skenario yang telah dikembangkan diaplikasikan dengan program aplikasi. Setiap sub-skenario direplikasi sebanyak 10 kali. Adanya keterbatasan pada program aplikasi maka jumlah *job* yang diujikan hanya dari jumlah *job* 2 sampai dengan 9 dan jumlah operasi dari 3 sampai dengan 5.

Tabel 4.1 Rekap hasil replikasi skenario

Job	Skenario	Rate	Biaya															
			1			2			3			4			5			
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
3	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
4	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
5	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
6	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
7	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
8	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
10	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
11	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
12	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
13	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
14	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
15	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
16	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
17	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
18	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
19	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
20	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
21	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
22	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
23	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
24	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
25	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
26	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
27	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
28	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
29	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
30	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9	0,05	0,25	0,5	0,7	0,8	0,9
Rata-rata																		

\* biaya rata-rata dalam ratusan Rupiah

Dari proses simulasi kita dapat melihat solusi yang dihasilkan oleh masing-masing algoritma. Pada satu kali simulasi dilakukan sebanyak 100 iterasi. Pada setiap algoritma, solusi yang terbaik dalam setiap simulasi dicapai pada urutan iterasi yang berbeda. Sebagai contoh pada simulasi *job* 6 operasi 4 dilakukan sebanyak 30 kali simulasi dan pada tiap simulasi dilakukan 100 iterasi. Solusi terbaik dalam setiap simulasi ditabelkan pada tabel berikut:

Tabel 4.2 Titik kemunculan solusi terbaik dalam simulasi

Simulasi ke-	Algoritma Tabu mencapai solusi terbaik pada waktu ke-	Algoritma Genetik mencapai solusi terbaik pada iterasi ke-
1	72	59
2	42	28
3	57	23
4	55	3
5	5	16
6	24	39
7	52	32
8	45	7
9	62	27
10	24	25
11	42	4
12	32	79
13	10	17
14	22	5
15	52	45
16	81	17
17	42	33
18	14	4
19	74	46
20	81	24
21	97	50
22	17	1
23	75	38
24	5	46
25	67	18
26	77	56
27	65	5
28	41	11
29	31	22
30	62	35
Rata-rata	51,03333333	35,55666667

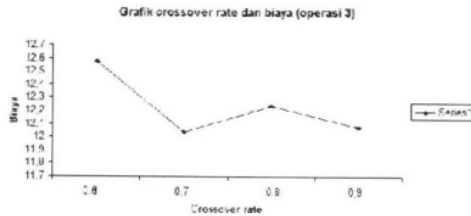
B. Analisa Data

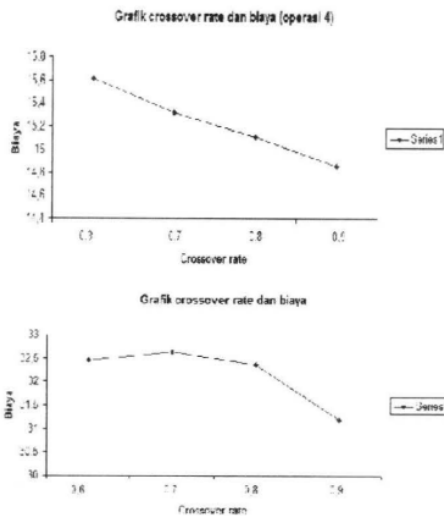
1). Analisa Parameter Algoritma Genetik

Pada tabel 4.1 tercantum parameter-parameter yang menghasilkan hasil terbaik dari algoritma Genetik. Parameter dalam algoritma Genetik yaitu *crossover rate* dan *mutation rate*. Hasil replikasi skenario menunjukkan tidak ada pasangan parameter tertentu yang selalu menghasilkan biaya terendah.

Pada parameter *crossover rate* biaya terendah dihasilkan hanya satu kali oleh *rate* 0,6 dan secara merata tersebar pada *rate* 0,7 yaitu sebanyak 6 sub-skenario, *rate* 0,8 yaitu sebanyak 5 sub-skenario, *rate* 0,9 yaitu sebanyak 8 sub-skenario.

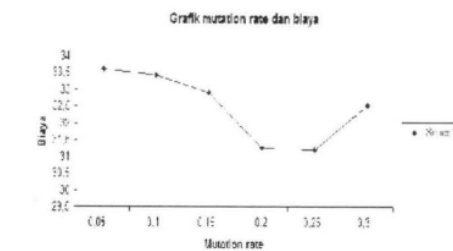
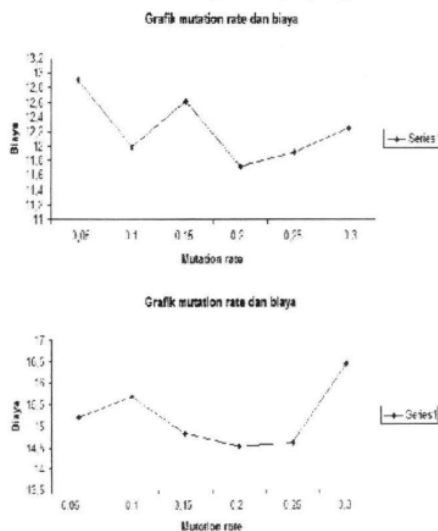
Pada parameter *mutation rate* biaya terendah dihasilkan dua sub-skenario oleh *rate* 0,05 dan 0,15, sebanyak 9 sub-skenario oleh *rate* 0,2 dan sebanyak 7 sub-skenario oleh *rate* 0,25. Berikut merupakan grafik antara *crossover rate* dan *mutation rate* terhadap biaya rata-rata pada masing-masing operasi.





Gambar 5.1 Grafik *crossover rate* dan biaya rata-rata

Melalui uji coba yang telah dilakukan maka dapat dilihat bahwa *crossover rate* menghasilkan biaya terendah saat *crossover rate* cukup tinggi, yaitu pada rentang 0,7; 0,8 dan 0,9. Hal ini disebabkan karena semakin besar *crossover rate* maka semakin besar pula peluang terbentuknya individu baru dalam populasi sehingga juga memungkinkan terbentuknya solusi yang lebih baik. Hal inilah yang menjadi penyebab bahwa biaya terendah dihasilkan saat *crossover rate* cukup tinggi yaitu pada rate 0,7; 0,8 dan 0,9. Dari grafik terlihat bahwa semakin besar *crossover rate* maka semakin kecil pula biaya yang timbul.



Gambar 5.2 Grafik *mutation rate* dan biaya rata-rata

Melalui uji coba yang telah dilakukan maka dapat dilihat bahwa *mutation rate* menghasilkan biaya terendah saat *mutation rate* 0,2 dan 0,25. Hal ini disebabkan karena mutasi memungkinkan terbentuknya individu dengan gen yang tidak ada dalam populasi awal sehingga proses mutasi akan mengakomodasi terbentuknya solusi yang lebih baik. Hal ini yang menjadi penyebab bahwa biaya terendah bukan hanya dihasilkan saat *crossover rate* yang tinggi tetapi juga saat *mutation rate* cukup tinggi. Akan tetapi dari grafik terlihat bahwa kenaikan *mutation rate* tidak selalu diikuti dengan penurunan biaya, jadi hanya pada kasus ini *mutation rate* 0,2 dan 0,25 menghasilkan biaya terendah. Secara alami maupun buatan mutasi adalah proses yang jarang terjadi sehingga *mutation rate* dibuat lebih kecil dibanding dengan *crossover rate*.

## 2). Analisa Parameter Algoritma *Tabu Search*

Pada tabel 4.1 juga tercantum parameter yang menghasilkan hasil terbaik dari algoritma *Tabu Search*. Parameter dalam algoritma *Tabu Search* yaitu *tabu list*. Seperti halnya dalam algoritma Genetik, hasil replikasi pada algoritma *Tabu Search* juga tidak menunjukkan ukuran parameter *Tabu Search* tertentu yang selalu menghasilkan biaya terendah.

Parameter *tabu list* yang diujikan mulai 2 sampai dengan 7. Ukuran *tabu list* 2 menghasilkan biaya terendah sebanyak 2 sub-skenario. Ukuran *tabu list* 3 menghasilkan biaya terendah sebanyak 1 sub-skenario. Ukuran *tabu list* 4 menghasilkan biaya terendah sebanyak 1 sub-skenario. Ukuran *tabu list* 5 menghasilkan biaya terendah sebanyak 1 sub-skenario. Ukuran *tabu list* 6 menghasilkan biaya terendah sebanyak 1 sub-skenario. Ukuran *tabu list* 7 menghasilkan biaya terendah sebanyak 1 sub-skenario. Ukuran *tabu list* 8 menghasilkan biaya terendah sebanyak 2 sub-skenario. Ukuran *tabu list* 9 menghasilkan biaya terendah sebanyak 10 sub-skenario. Ukuran *tabu list* 10 tidak menghasilkan biaya terendah dan ukuran *tabu list* 11 menghasilkan biaya terendah sebanyak 1 sub-skenario.

Hasil tersebut menunjukkan biaya terendah dihasilkan paling banyak oleh ukuran *tabu list* 9. Berikut merupakan grafik antara ukuran *tabu list* dan biaya rata-rata pada masing-masing operasi.

#### IV. KESIMPULAN DAN SARAN

##### Kesimpulan

1. Parameter algoritma Genetik menghasilkan biaya terendah pada saat *crossover rate* 0,7; 0,8; 0,9 dan *mutation rate* 0,2 dan 0,25.
2. Parameter algoritma *Tabu Search* menghasilkan biaya terendah pada saat ukuran *tabu list* 9.
3. Dari uji coba yang telah dilakukan, algoritma *Tabu Search* menghasilkan biaya yang lebih rendah daripada algoritma Genetik pada kasus *job shop* dengan mesin paralel dan kriteria minimasi biaya *earliness* dan *tardiness*.

#### DAFTAR PUSTAKA

- [1]. Baker, K.R (1974) *Introduction to Sequencing and Scheduling*. John Wiley & Sons. New York.
- [2]. Bedworth, D. dan Bailey, J.E. (1987) *Integrated Production Control Systems*. John Wiley & Sons, Inc. Singapore.
- [3]. Dell'Amico, M dan Trubian, M. (1993) *Applying Tabu Search to The Job-Shop Scheduling Problem*. *Annals of Operations Research*, 41, 231-252.
- [4]. Fogarty, D.W, Blackstone, J.H and Hoffman, Thomas R. (1991) "*Production and Inventory Management*", Cincinnati, Ohio, South-Western Publishing Co.,
- [5]. Gen, M. dan Cheng, R. (1997). *Genetic Algorithms & Engineering Design*. John Wiley & Sons, Inc. New York.
- [6]. Goldberg, D.E. (1989) *Genetic Algorithm in Search, Optimization, & Machine Learning*. Addison-Wesley Publishing Company, Inc. Massachusetts.
- [7]. Morton, T.E. dan Pentico, D.W. (1993) *Heuristic Scheduling Systems with Application to Production System and Project Management*. John Wiley & Sons, Inc. Canada.



# Perbandingan Algoritma Tahu Search dan Algoritma .. Genetika dalam Penjadwalan Mesin untuk Minimasi Biaya Earliness dan Tardiness

## ORIGINALITY REPORT

10%

SIMILARITY INDEX

10%

INTERNET SOURCES

2%

PUBLICATIONS

1%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://journal.wima.ac.id">journal.wima.ac.id</a> Internet Source	3%
2	<a href="http://repository.uki.ac.id">repository.uki.ac.id</a> Internet Source	1%
3	<a href="http://repository.uin-suska.ac.id">repository.uin-suska.ac.id</a> Internet Source	1%
4	(2-15-14) <a href="http://202.46.29.77/images/kumpulan%20abstrak%20202">http://202.46.29.77/images/kumpulan%20abstrak%202</a> Internet Source	1%
5	<a href="http://docplayer.info">docplayer.info</a> Internet Source	1%
6	<a href="http://pt.scribd.com">pt.scribd.com</a> Internet Source	1%
7	<a href="http://repository.usd.ac.id">repository.usd.ac.id</a> Internet Source	1%
8	<a href="http://www.tc.faa.gov">www.tc.faa.gov</a> Internet Source	1%

9

Ramirez Narvaez Maria Eugenia, Gonzalez  
Ávila Miguel Angel, Garcia Duarte Jesus  
Antonio. "Estudio descriptivo de redes de  
area local", TESIUNAM, 1991

Publication

1 %

---

Exclude quotes      On

Exclude matches      < 1%

Exclude bibliography      On