

Implementasi Algoritma Genetik-Tabu Search dalam Optimasi Penjadwalan Perkuliahan

Rusianah, M. Aziz Muslim, Sholeh Hadi Pramono

Abstract - Schedule is important, since implementation of course involves many aspects and each has different priorities. Scheduling is a complicated job, in addition to put properly existing components such as faculty, courses, lectures time and space, it should also meet the constraints. During this time at the Department of Informatics Engineering, scheduling is done manually, causing problems such as long processing time and the result is also far to perfect because violation of scheduling rules exist. The duration of the scheduling process also affects other academic processes, such as programming courses next semester. This scheduling problem can be solved by optimization. The algorithm used in this research are Genetic Algorithm and Tabu Search. In this study, comparison of Genetic Algorithm (GA) with hybridization of Genetic Algorithm -Tabu Search (GA-TS) is made. Comparison results show that GA-TS has better performance than GA, in term of fitness and convergence time.

Index : Class Scheduling, Genetic Algorithms, Tabu Search

Abstrak – Penjadwalan perkuliahan merupakan hal penting, karena pelaksanaan kegiatan perkuliahan melibatkan banyak aspek yang berkaitan dan masing-masing memiliki prioritas yang berbeda. Penjadwalan ini merupakan pekerjaan yang rumit. Selain harus bisa menempatkan komponen yang ada seperti dosen, mata kuliah, ruang dan waktu perkuliahan dengan tepat, juga harus bisa memenuhi aturan (*constraints*) yang sudah ditentukan. Selama ini di Jurusan Teknik Informatika, penjadwalan perkuliahan dilakukan secara manual, sehingga menimbulkan permasalahan seperti proses pengerjaan yang lama dan hasilnya juga kurang sempurna karena masih adanya pelanggaran terhadap aturan penjadwalan. Lamanya proses penjadwalan juga mempengaruhi proses akademik yang lain, seperti pemrograman mata kuliah pada semester berikutnya. Masalah penjadwalan ini bisa diselesaikan dengan optimasi. Algoritma yang dipakai dalam penelitian ini adalah Algoritma Genetika dan *Tabu Search*. Dalam penelitian ini akan dikomparasikan hasil dari penggunaan metode Algoritma Genetika (AG) dengan penggabungan Algoritma Genetik-*Tabu Search* (AG-TS). Hasil komparasi menunjukkan bahwa AG-TS mempunyai performa yang lebih baik dari pada AG, dalam hal *fitness* maupun konvergensi waktu.

Kata kunci : *Penjadwalan Perkuliahan, Algoritma Genetika, Tabu Search*

Rusianah, M. Aziz Muslim, Sholeh Hadi Pramono are with the Electrical Engineering Department of Brawijaya University, Malang, Indonesia (corresponding author provide phone 0341-554166 (ocis_ana@yahoo.com))

I. PENDAHULUAN

PROSES penjadwalan perkuliahan merupakan salah satu tahapan yang memerlukan proses pengaturan dalam kegiatan akademik di suatu perguruan tinggi. Penjadwalan perkuliahan merupakan hal penting, karena pelaksanaan kegiatan perkuliahan melibatkan banyak aspek yang berkaitan dan masing-masing memiliki prioritas yang berbeda. Aspek-aspek yang harus dipertimbangkan dalam penjadwalan diantaranya adalah: jumlah mahasiswa, dosen, waktu, ruangan, serta persyaratan lainnya. Masalah penjadwalan kuliah akan berbeda untuk universitas yang berbeda [1]. Artinya bahwa setiap lembaga pendidikan dalam hal ini universitas, mempunyai masalah penjadwalan yang tidak sama.

Pekerjaan penjadwalan ini merupakan pekerjaan yang rumit. Selain harus bisa menempatkan komponen yang ada seperti dosen, mata kuliah, ruang dan waktu perkuliahan dengan tepat, juga harus bisa memenuhi aturan (*constraint*) penjadwalan yang sudah ditentukan. Berikut beberapa aturan yang harus dipenuhi:

- Tidak ada dosen yang mengajar lebih dari satu kelas mata kuliah yang berbeda pada waktu yang sama.
- Lebih dari satu kelas mata kuliah yang berbeda tidak dapat berada pada ruang dan waktu yang sama. Tidak mungkin ada dua kelas mata kuliah yang berbeda diselenggarakan bersamaan di satu ruangan.
- Satu Dosen tidak boleh mengajar lebih dari 2 sks berturut-turut/harus ada jeda waktu istirahat.
- Mahasiswa pada semester yang sama tidak boleh mengikuti perkuliahan 2 jadwal matakuliah berturut-turut.

Selama ini di Jurusan Teknik Informatika penjadwalan perkuliahan dilakukan secara manual, sehingga menimbulkan permasalahan seperti proses pengerjaan yang lama dan hasilnya juga kurang sempurna karena masih adanya pelanggaran terhadap aturan penjadwalan. Lamanya proses penjadwalan juga mempengaruhi proses akademik yang lain, seperti pemrograman mata kuliah pada semester berikutnya. Masalah penjadwalan ini bisa diselesaikan dengan optimasi. Dalam penelitian ini, akan menggunakan Algoritma Genetika (AG) dan *Tabu Search* (TS) sebagai metode optimasi dalam aplikasi penjadwalan perkuliahan.

Algoritma Genetika (AG) adalah salah satu metode *metaheuristik* dalam kelompok *Evolutionary Algorithm* (EA) yaitu algoritma pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses

biologi dari evolusi. AG bisa dimodifikasi secara sangat bebas sehingga sesuai dengan masalah yang dihadapi [2]. Keuntungan penggunaan AG adalah kemudahan implementasi dan kemampuannya untuk menemukan solusi yang tepat untuk masalah-masalah berdimensi tinggi [3]. *Tabu Search* (TS) adalah suatu metode optimasi matematis yang termasuk kedalam kelas *local search*. TS memperbaiki performansi *local search* dengan memanfaatkan penggunaan struktur *memory*. Karena itu, untuk mengoptimalkan proses penjadwalan perkuliahan dalam penelitian ini, maka digunakanlah penggabungan algoritma genetika dan *tabu search* guna mendapatkan hasil penjadwalan yang tepat dengan waktu yang cepat (minimal). Hasil dari penggabungan AG-TS ini nantinya akan dibandingkan dengan yang hanya menggunakan AG saja. Perbandingan dilakukan untuk komparasi ketepatan dan kecepatan dari keduanya dalam proses penjadwalan. Output yang dihasilkan berupa susunan jadwal perkuliahan yang utuh seperti pada Tabel I.

II. TINJAUAN PUSTAKA

A. Penjadwalan Perkuliahan

Penjadwalan adalah penempatan sumber daya (*resource*) dalam satu waktu. Penjadwalan mata kuliah merupakan persoalan penjadwalan yang umum dan sulit dimana tujuannya adalah menjadwalkan pertemuan dari sumber daya. Sumber daya yang dimaksud adalah dosen pengasuh mata kuliah, mata kuliah, ruang kuliah, kelas mahasiswa, dan waktu [4].

Constraint

Terdapat batasan-batasan (*constraints*) dalam penyusunan penjadwalan mata kuliah. Batasan-batasan dalam penjadwalan dibagi ke dalam dua kategori yaitu “*hard*” dan “*soft*” *constraints*. *Constraint* penjadwalan perkuliahan yang berlaku tentunya berbeda tergantung dari aturan dan tata cara perkuliahan dari institusi yang bersangkutan. *Hard constraints* adalah batasan yang

1. Mata kuliah yang berbeda tidak boleh dijadwalkan pada waktu dan tempat yang sama (*hard constraint*).
2. Dalam satu hari mahasiswa tidak boleh menghadiri lebih dari 3x *timeslot* kuliah berturut-turut, harus ada jeda minimal 1 *timeslot* (*soft constraint*).

Jadwal untuk perkuliahan biasanya mingguan dan berlaku untuk satu semester. Sebuah jadwal disebut lengkap jika setiap mata kuliah yang ditawarkan pada suatu semester memiliki slot kapan kuliah tersebut berlangsung, ruang berlangsung dan dosen yang mengajar. Jadwal yang lengkap belum tentu valid. Jadwal yang valid adalah jadwal yang sesuai dengan *constraint* yang harus dipenuhi agar kegiatan perkuliahan dapat benar-benar berlangsung.

Optimasi

Pengertian optimasi bisa dijelaskan sebagai suatu kumpulan formula matematis dan metoda numerik untuk menemukan dan mengidentifikasi kandidat terbaik dari sekumpulan alternatif tanpa harus secara eksplisit menghitung dan mengevaluasi semua alternatif yang mungkin [5]. Banyak kasus atau masalah dalam kehidupan sehari-hari yang melibatkan optimasi untuk memecahkannya. Contoh beberapa kasus optimasi diantaranya adalah masalah penjadwalan, *crew scheduling*, *Traveling Salesman Problem* (TSP), *Capacitated Vehicle Routing* (CVRP), dan lain sebagainya. Optimasi memegang peran penting dalam proses mendesain suatu sistem. Dengan optimasi, desain suatu sistem bisa menghasilkan ongkos yang lebih murah atau profit yang lebih tinggi, menurunkan waktu proses dan sebagainya.

B. Algoritma Genetika

Algoritma Genetika (AG) adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma genetika banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan lainnya.

TABEL I.
JADWAL PERKULIAHAN

No.	Hari	Jam	Kode MK	Mata Kuliah	Kelas	SKS	SMT	Dosen	Ruang
1	Senin	06.30 - 07.20	0765315	SISTEM INFORMASI CERDAS	B	3	7	LINDA SALMA ANGREANI, M.T	B306
2	Senin	08.10 - 09.00	0765301	KECERDASAN BUATAN	B	3	5	HANI NURHAYATI, M.T	B316
3	Senin	09.00 - 09.50	0765402	ISU PROFESIONAL DAN PROYEK GRUP	D	2	7	IR. JASMANI, M.KOM	B324
4	Senin	09.30 - 10.40	0765110	STUDI FIQH	C	2	6	ABD. ROZAQ, M.AG RIRIEN	B306
5	Senin	09.00 - 09.50	0765203	ALJABAR LINIER DAN MATRIKS	D	3	3	KUSUMAWATI, M.KOM	B316
.....

wajib untuk dipenuhi atau tidak boleh dilanggar. Sedangkan *soft constraints* adalah batasan yang masih member toleransi terhadap pelanggaran, namun sebisa mungkin diminimalisir pelanggarannya. Contoh *hard constraint* dan *soft constraint* pada penjadwalan perkuliahan adalah:

Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi. Solusi-solusi dari sebuah populasi diambil dan digunakan untuk membentuk populasi yang baru. Hal ini dimotivasi dengan harapan bahwa populasi yang baru dibentuk tersebut akan lebih baik daripada yang lama. Solusi-solusi yang dipilih untuk membentuk solusi-solusi yang baru dipilih sesuai

dengan *fitness* mereka masing-masing [6]. Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai *fitness* ini yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma Genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi.

Secara umum sebuah penerapan algoritma genetika akan melalui siklus sederhana yang terdiri dari empat langkah, yaitu [7] :

- a. Membangun sebuah “populasi” yang terdiri dari beberapa *string*.
- b. Evaluasi masing-masing *string* (*fitness value*).
- c. Proses seleksi agar didapat *string* yang terbaik.
- d. Manipulasi genetika untuk menciptakan populasi baru dari *string*.

C. Tabu Search

Tabu Search (TS) adalah suatu metode optimasi yang menerapkan sistem pemanfaatan memory. Pada metode *Tabu Search*, solusi yang sudah pernah ditemui akan dilarang untuk di eksekusi kembali pada iterasi berikutnya. *TS* menggunakan struktur *memory* yang disebut *Tabu List*. *Tabu List* digunakan untuk menyimpan solusi yang pernah ditemui selama iterasi berjalan agar proses pencarian tidak berulang-ulang pada daerah solusi yang sama, selain itu untuk menuntun proses pencarian menelusuri solusi-solusi yang belum pernah dikunjungi sebelumnya.

Secara umum, prosedur *TS* dapat dituliskan sebagai berikut [2] :

1. Bangkitkan solusi awal yang layak, misalkan s , secara acak atau menggunakan metode heuristik tertentu.
2. $BiayaOptimum = Biaya(s)$
3. $s^* = s$ { s^* adalah solusi terbaik yang diperoleh}
4. $TabuList = null$
5. *repeat*
 - a. $V^* =$ himpunan solusi yang merupakan tetangga dari s yang memenuhi criteria aspirasi atau tidak berada dalam *TabuList*
 - b. Pilih s' { s' adalah solusi yang memiliki biaya minimum di dalam V^* }
 - c. Simpan *move* yang berlawanan kedalam *TabuList*, yang mengubah s ke s'
 - d. $s = s'$
if ($Biaya(s) < BiayaOptimum$) then
 $s^* = s$
 $BiayaOptimum = Biaya(s)$
End
 Until (KriteriaBerhenti = true)
6. return (s^*)

III. METODE PENELITIAN

Penerapan referensi terkait penelitian diperlukan untuk keberhasilan pencapaian tujuan. Sehingga perlu dijabarkan mengenai proses dan pengujian sistem, untuk lebih jelasnya sebagai berikut :

A. Konsep Algoritma

Pada Penelitian ini akan digunakan dua metoda *metaheuristik* yaitu AG dan TS dalam penyusunan Jadwal Perkuliahan. Dalam hal ini penggunaan AG akan digabungkan dengan TS. AG disini menggunakan pengkodean bilangan integer dan melakukan *crossover* (perkawinan silang) dengan persilangan dua titik. Pada proses seleksi dalam penelitian ini akan menggunakan TS sebagai operatornya.

Selain itu dalam penelitian ini juga menggunakan TS dalam proses *crossover* seperti pada penelitian yang telah dilakukan oleh Setemen dan Purnomo [8], dimana dalam penelitiannya TS digunakan sebagai filter kromosom yang telah mengalami *crossover* agar kromosom yang sama tidak dilakukan *crossover* berulang-ulang. Pada iterasi pertama semua kromosom yang telah dilakukan *crossover* akan disimpan dalam *tabu list*, kemudian pada iterasi berikutnya kromosom yang mengalami *crossover* akan dicek dalam *tabu list* kalau sudah ada maka hanya perlu dilakukan mutasi saja. Selain itu, dalam penelitian ini akan dibandingkan antara menggunakan metode AG dengan kombinasi AG-TS.

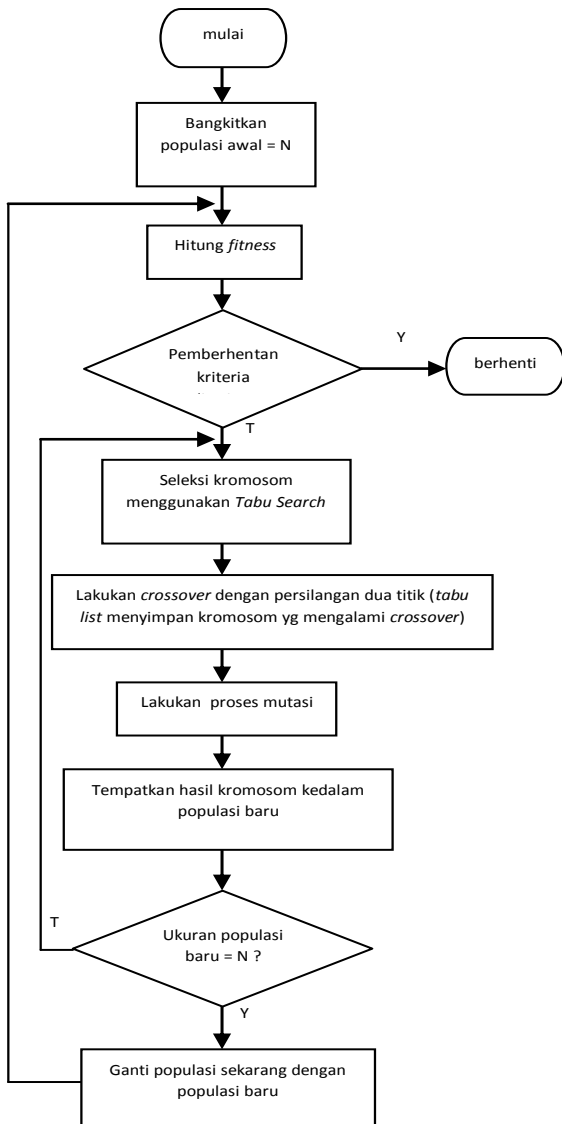
Flowchart penggabungan AG-TS dapat dilihat pada Gambar 1.

Representasi Kromosom

Algoritma Genetika tidak beroperasi dengan penyelesaian asli dari suatu masalah tetapi beroperasi dengan penyelesaian yang telah direpresentasikan. Representasi kromosom merupakan proses pengkodean dari penyelesaian asli dari suatu permasalahan. Pengkodean kandidat penyelesaian ini disebut dengan kromosom.

Kromosom dalam penjadwalan ini berupa satu slot jadwal utuh yang berisi komponen hari, jam, mata kuliah, dosen pengampu, kelas dan ruang perkuliahan. Komponen pembentuk kromosom jadwal itu disebut sebagai gen.

Dalam penelitian mengenai penjadwalan perkuliahan ini berkaitan dengan komponen yang digunakan seperti hari sebanyak 6, jam kuliah sebanyak 13 slot, mata kuliah sebanyak 29 MK, dosen sebanyak 32 orang, ruang yang dipakai sebanyak 6 ruang, dan kelas sebanyak 4 kelas maka akan dirubah dalam bentuk bilangan integer.



Gambar 1. Flowchart Penggabungan AG-TS

Fungsi Fitness

Fungsi *fitness* digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan. Fungsi *fitness* tersebut sebagai berikut:

$$Fitness = \frac{1}{1+penalty} \tag{1}$$

Dari persamaan diatas nilai *fitness* ditentukan oleh nilai *penalty*. *Penalty* tersebut menunjukkan jumlah pelanggaran kendala pada suatu kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai *penalty* berbanding terbalik dengan nilai *fitness*, semakin kecil nilai *penalty* (jumlah pelanggaran) semakin besar nilai *fitness*nya.

Didalam penjadwalan perkuliahan ini *penalty* menunjukkan besarnya nilai pelanggaran terhadap *constraints*, yaitu pelanggaran terhadap *hard constraints* maupun *soft constraints*. Bobot nilai pelanggaran

TABEL II.
PERUBAHAN DATA DALAM BILANGAN INTEGER

Hari	integer
Senin	1
Selasa	2
Rabu	3
Kamis	4
Jum'at	5
Sabtu	6

terhadap *hard constraint* adalah 1 dan pelanggaran terhadap *soft constraint* adalah 0,5. Maka model perhitungannya adalah sebagai berikut;

$$Fitness = 1/(1+jumlah\ penalty\ hard\ constraint + penalty\ soft\ constraint)$$

B. Metode Pengujian

Pengujian dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Dalam hal ini akan dilakukan pengujian dengan membandingkan hasil dari penjadwalan perkuliahan yang menggunakan metode Algoritma Genetika saja dengan yang menggunakan penggabungan Algoritma Genetik-*Tabu Search*. Yaitu dengan melihat *fitness*, waktu proses, dan konvergen. Konvergen disini maksudnya adalah generasi yang mempunyai nilai *fitness* tertinggi.

C. Penetapan Variabel

Berdasarkan permasalahan dan tinjauan pustaka yang relevan dengan topic penelitian ini maka didapatkan variabel yang akan digunakan. Secara garis besar data variabel yang digunakan dalam penelitian ini adalah :

- Dosen Pengampu dan Matakuliah
- Waktu Perkuliahan
- Ruang Kuliah
- Kelas Kuliah

IV. PENGUJIAN DAN PEMBAHASAN

A. Persiapan Data

Pada tahap pengujian ini, data yang digunakan adalah data perkuliahan regular semester ganjil tahun akademik 2014/2015 pada Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas X di Malang. Data yang digunakan dapat dilihat pada Tabel III.

B. Pengujian

Pengujian ini bertujuan untuk mengetahui unjuk kerja dari sistem aplikasi yang menggunakan AG dan yang menggunakan penggabungan AG-TS. Dari pengujian ini akan diketahui *fitness* yang diperoleh, waktu proses yang dibutuhkan dan pada generasi keberapa hasil yang konvergen didapatkan. Percobaan pada masing-masing metode menggunakan parameter yang sama yaitu dengan maksimum iterasi sebanyak 1000, Probabilitas mutasi (*Pm*) 0.1, sedangkan untuk Probabilitas *crossover* (*Pc*) ada tiga 0.6, 0.7,0.8. berikut hasil dari percobaan yang dilakukan:

TABEL III.
DATA YANG DIGUNAKAN

No.	Komponen	Jumlah
1	Hari	6 Hari
2	Waktu Perkuliahan	13 slot
3	Mata Kuliah	29 MK
4	Dosen	32 Orang
5	Ruangan	6 ruang
6	Kelas	4 kelas

Satuan waktu yang digunakan 50 menit per jam kuliah yang berbobot 1 SKS.

TABEL IV.
PERCOBAAN AG-TS

Pc	P	F	W	G
0.6	0	1	59	569
0.7	0	1	57	571
0.8	1	0,5	100	1000

TABEL V.
PERCOBAAN AG

Pc	P	F	W	G
0.6	0	1	70	754
0.7	1	0.5	91	1000
0.8	0	1	107	987

Keterangan

- Pc = Probabilitas *crossover*
 P = Pinalti (nilai *constraint* terlanggar)
 F = *Fitness*
 W = Waktu proses (detik)
 G = Generasi

C. Analisis

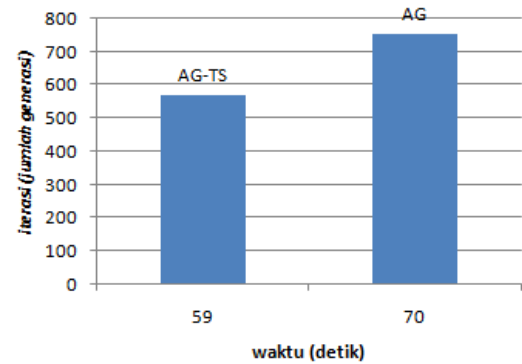
Uji coba bertujuan membandingkan performa dari sistem aplikasi yang menggunakan AG dengan yang menggunakan AG-TS. Perbandingan dilakukan pada hasil uji yang konvergen yaitu mempunyai nilai *fitness* = 1, yang dibandingkan adalah waktu proses dan generasi. Hal tersebut dapat dilihat pada Gambar 2 dan Gambar 3.

Konvergensi *fitness* = 1 berarti bahwa dalam penjadwalan itu tidak terdapat pelanggaran terhadap *hard constraint* maupun *soft constraint*.

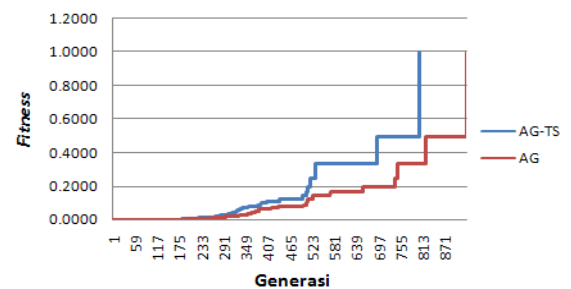
Berikut ini juga dilakukan percobaan untuk mengetahui rekaman nilai *fitness* hingga mencapai nilai *fitness* = 1. Percobaan dilakukan dengan menggunakan populasi = 10 individu, probabilitas *crossover* = 0.6, probabilitas mutasi = 0.1. Pada Gambar 3 dibawah ini dapat dilihat rekaman nilai *fitness* dalam bentuk grafik.

Berdasarkan grafik pada Gambar 3, masing-masing metode yang digunakan menunjukkan perbaikan nilai *fitness* pada setiap generasi. Pada penggunaan metode AG-TS lebih cepat mencapai konvergen bila dibandingkan dengan penggunaan metode AG saja.

Dari beberapa percobaan diatas, dapat ditarik kesimpulan bahwa dengan penggunaan metode AG-TS lebih cepat mencapai nilai optimal dibanding metode AG. Hal ini menunjukkan bahwa performansi dari penggunaan metode AG-TS lebih baik bila dibandingkan dengan penggunaan metode AG.



Gambar 2. Grafik Komparasi Algoritma



Gambar 3. Grafik Komparasi Rekaman Nilai Fitness

V. KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Hasil implementasi yang diujikan dalam sistem aplikasi penjadwalan perkuliahan menggunakan AG-TS menghasilkan performa yang tinggi dengan konvergen *fitness* = 1 pada parameter probabilitas *crossover* 0.6, waktu proses selama 59 detik di generasi ke-569. Sedangkan untuk penggunaan AG menghasilkan performa yang tinggi dengan konvergen *fitness* = 1 pada parameter probabilitas *crossover* 0.6, waktu proses selama 70 detik di generasi ke-754. *Fitness* = 1 artinya bahwa jadwal yang dihasilkan sudah memenuhi kriteria persyaratan (*constraint*) yang ada baik itu *hard constraint* maupun *soft constraint*. Dengan kata lain sudah didapatkan jadwal yang tepat. Probabilitas *crossover* 0.6 artinya bahwa besarnya peluang untuk menghasilkan keturunan dengan proses *crossover* sebesar 60%. Waktu proses adalah waktu yang diperlukan dalam sekali percobaan menjalankan sistem aplikasi. Diketahui waktu proses menggunakan AG-TS selama 59 detik dan

AG selama 70 detik, yang artinya waktu proses AG-TS untuk mencapai konvergen $fitness = 1$ lebih cepat dari pada AG. Generasi menunjukkan banyaknya iterasi yang dibutuhkan untuk mencapai konvergen. Dilihat dari hasil yang sudah diperoleh, AG-TS konvergen pada generasi ke-569 dan AG pada generasi ke-754. Hal itu dapat dikatakan bahwa iterasi untuk mencapai konvergen dengan AG-TS lebih sedikit dari pada AG.

2. Sehingga dalam penelitian ini jika dilihat dari hasil uji coba tersebut, dapat disimpulkan bahwa penggunaan AG-TS memiliki unjuk kerja yang lebih baik dibandingkan penggunaan AG saja.

REFERENSI

- [1] Ruey-Maw Chen and Hsiao-Fang Shih. 2013. *Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search*. Algorithms. 6: 227-244. www.mdpi.com/journal/algorithms.
- [2] Suyanto. 2010. *Algoritma Optimasi Deterministik dan Probabilistik*. Graha Ilmu. Yogyakarta.
- [3] Suyanto. 2005. *Artificial Intelligence Searching-Reasoning-Planning-Learning*. Informatika. Bandung.
- [4] Edward L. Mooney, dkk. 1995. *LARGE SCALE CLASSROOM SCHEDULING*. Industrial and Management Engineering Department. Montana State University.
- [5] Willy, Paul dan Santosa, Budi. 2011. *Metoda Metaheuristik Konsep dan Implementasi*. Guna Widya. Surabaya.
- [6] Juniawati. 2003. *Implementasi Algoritma Genetika untuk Mencari Volume Terbesar Bangun Kotak Tanpa Tutup dari Suatu Bidang Datar Segi Empat*. Jurnal Ilmiah LPPM Universitas Surabaya. Surabaya.
- [7] Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.
- [8] Setemen, Komang dan Purnomo, Mauridhi, Hery. 2008. *Kombinasi Algoritma Genetika dan Tabu Search dalam Pembuatan Tabel Jadwal Mata Kuliah*. Seminar on Intelligent Technology and Its Applications. 375-378.