

## Optimasi Penjadwalan Mata Pelajaran Menggunakan Algoritme Genetika (Studi Kasus: SMK Negeri 2 Kediri)

Muhammad Fuad Efendi<sup>1</sup>, Imam Cholissodin<sup>2</sup>, Edy Santoso<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>muhfuadeffendi@gmail.com, <sup>2</sup>imamcs@ub.ac.id, <sup>3</sup>edy144@ub.ac.id.

### Abstrak

Proses penyusunan jadwal secara manual dirasa kurang efisien karena membutuhkan waktu yang lama. Permasalahan penyusunan jadwal akan kompleks apabila jumlah komponen yang semakin banyak atau jumlah data yang besar dari setiap komponen tersebut. Hasil jadwal yang diharapkan bukan hanya jadwal yang tidak mengalami bentrok, akan tetapi jadwal yang dapat menyesuaikan terhadap beberapa *constraint* yang harus terpenuhi di dalam jadwal tersebut. Algoritme Genetika merupakan algoritme yang bersifat iteratif, menyesuaikan diri dan algoritme probabilistik dalam pencarian untuk optimasi global. Proses inialisasi kromosom yang dibangkitkan dari data penugasan guru dengan representasi bilangan *integer* yang setiap gen berisi kode-kode penugasan yang telah dibangkitkan secara *random*. Setiap kromosom dengan nilai *fitness* tertinggi merupakan representasi dari solusi jadwal mata pelajaran. Dari proses pengujian yang telah dilakukan didapatkan beberapa nilai parameter-parameter Algoritme Genetika yaitu nilai jumlah populasi terbaik adalah 90, nilai kombinasi *Cr* dan *Mr* adalah 0.5 dan 0.5, dan jumlah generasi sebanyak 40000. Proses pencarian solusi dengan menggunakan parameter-parameter tersebut didapatkan nilai *fitness* yaitu sebesar 0,8451.

**Kata Kunci:** *algoritme genetika, penjadwalan, mata pelajaran.*

### Abstract

*The process of drafting schedules manually felt less efficient because it takes a long time. The problem of drafting the schedule will be complex if the number of components is more large amount of data from each component. The expected schedule is not just a schedule that does not clash, but a schedule that can adapt to some constraints that must be met within the schedule. Genetic Algorithms are algorithms that are iterative, self-adjusting and probabilistic algorithms in search for global optimization. The process of chromosome initialization generated from teacher assignment data by integer representation of each gene containing randomly generated assignment codes. Each chromosome with the highest fitness value is a representation of the subject schedule solution. From the testing process that has been done, has obtained the parameters of Genetic Algorithm is the best population number is 90, the value of the combination of Cr and Mr is 0.5 and 0.5, and the number of generations as much as 40000. The process of finding solutions using these parameters obtained the value of fitness that is 0,8451.*

**Keywords:** *genetic algorithms, scheduling, school subjects*

## 1. PENDAHULUAN

Jadwal merupakan suatu hal yang harus ada dalam setiap kegiatan, salah satu kegiatan yang membutuhkan jadwal adalah kegiatan di dalam dunia pendidikan yaitu berupa jadwal mata pelajaran yang berfungsi sebagai pedoman yang digunakan oleh guru, siswa maupun kepala sekolah dalam melaksanakan proses belajar mengajar. Pengertian dari istilah jadwal pelajaran adalah urutan-urutan pelajaran sebagai

pedoman yang harus diikuti dalam proses belajar mengajar (Mulyana, 2010).

SMK Negeri 2 Kediri merupakan sekolah kejuruan yang memiliki problematika penjadwalan yang unik yaitu memiliki 57 kelas, 11 ruang lab dan terdapat 7 jurusan kejuruan dengan memiliki 4 jenis mata pelajaran yaitu mata pelajaran normatif, adaptif, mulok (Muatan Lokal) dan mata pelajaran kejuruan. Dalam proses belajar mengajarnya SMK Negeri 2 Kediri memiliki total 52 Jam pelajaran setiap minggu untuk setiap kelas dengan total guru

mata pelajaran adalah 113 guru.

Dalam proses pembuatan jadwal dilakukan hanya menggunakan *tools Microsoft Excel* yang telah diprogram untuk mendeteksi jadwal yang bentrok dan kebutuhan porsi jam mengajar setiap guru dari jadwal yang diinputkan dan dibuat secara manual oleh bagian waka kurikulum karena dalam proses penyusunan jadwal mata pelajaran pada SMK Negeri 2 Kediri tidak bisa hanya menggunakan *software timetabling* karena tidak sesuai dengan kondisi-kondisi yang ada di SMK Negeri 2 Kediri.

Masalah penjadwalan seperti yang telah dijabarkan dapat dilakukan optimasi agar dapat menghasilkan solusi dengan cepat dan optimal sesuai dengan aturan-aturan maupun batasan yang diinginkan. Algoritme Genetika merupakan salah satu algoritme yang banyak digunakan dalam bidang kecerdasan buatan dalam teknik optimasi (Promcharoen, et al., 2009). Algoritme Genetika merupakan algoritme yang bersifat iteratif, menyesuaikan diri dan algoritme probabilistik dalam pencarian untuk optimasi global (Wen & Wen-jun, 2006).

Dengan adanya penelitian dalam optimasi penjadwalan mata pelajaran ini diharapkan dalam penelitian ini dapat menghasilkan solusi yang optimal sesuai dengan permasalahan serta *constraint-constraint* yang terdapat pada SMK Negeri 2 Kediri.

## 2. PENJADWALAN

Penjadwalan merupakan pengaturan waktu dari suatu kegiatan operasi yang mencakup kegiatan mengalokasikan fasilitas, peralatan, maupun tenaga kerja dan menentukan urutan pelaksanaan bagi suatu kegiatan (Herjanto, 2006). Jadwal adalah sebuah tatanan dalam sebuah pertemuan pada suatu waktu yang merupakan kombinasi dari beberapa sumber daya seperti ruang, orang, dll. (Jain, et al., 2010). Dalam suatu lembaga pendidikan, penjadwalan diperlukan yaitu untuk penjadwalan mengajar, pengalokasian ruang kelas, tenaga pengajar, staf administrasi dan pendaftaran penerimaan mahasiswa baru atau ujian. Keputusan yang dibuat dalam penjadwalan adalah waktu mulai, waktu selesai, serta urutan operasi suatu pekerjaan.

### 2.1 Penjadwalan Sekolah

Proses pembentukan jadwal memiliki beberapa tahapan yaitu penentuan standar isi KTSP oleh dinas pendidikan yang digunakan

sebagai acuan, penentuan struktur program oleh kepala program pada masing-masing jurusan, pembagian tugas jam mengajar guru, dan kemudian disusun jadwal mata pelajaran oleh bagian kurikulum berdasarkan penugasan jam mengajar yang telah dibuat. dalam pembuatan jadwal juga diperhatikan mata pelajaran yang memerlukan ruang lab sebagai penunjang kegiatan belajar mengajar. Dalam penggunaan jadwal penggunaan lab dibuat oleh bagian kepala program studi (kaprog) Teknik Komputer dan Informatika. Alokasi waktu yang digunakan dalam pembentukan jadwal ditunjukkan pada Tabel 1 sampai Tabel 3.

**Tabel 1.** Alokasi waktu untuk hari senin-kamis

Jam ke		Waktu
0	Doa	6:45:00 - 7:00:00
1		7:00:00 - 7:40:00
2		7:40:00 - 8:20:00
3		8:20:00 - 9:00:00
4		9:00:00 - 9:40:00
	ISTIRAHAT1	9:40:00 - 10:00:00
5		10:00:00 - 10:40:00
6		10:40:00 - 11:20:00
7		11:20:00 - 12:00:00
	ISTIRAHAT2	12:00:00 - 12:40:00
8		12:40:00 - 13:20:00
9		13:20:00 - 14:00:00
10		14:00:00 - 14:40:00

**Tabel 2.** Alokasi waktu untuk hari jumat

Jam ke		Waktu
0	Doa	6:45:00 - 7:00:00
1	Jumat Bersih	7:00:00 - 7:40:00
2		7:40:00 - 8:20:00
3		8:20:00 - 9:00:00
4		9:00:00 - 9:40:00
	ISTIRAHAT1	9:40:00 - 10:00:00
5		10:00:00 - 10:40:00
6		10:40:00 - 11:20:00

**Tabel 3.** Alokasi waktu untuk hari sabtu

Jam ke		Waktu
0	Doa	6:45:00 - 7:00:00
1		7:00:00 - 7:40:00
2		7:40:00 - 8:20:00
3		8:20:00 - 9:00:00
4		9:00:00 - 9:40:00
	ISTIRAHAT1	9:40:00 - 10:00:00
5		10:00:00 - 10:40:00
6		10:40:00 - 11:20:00

## 3. ALGORITME GENETIKA

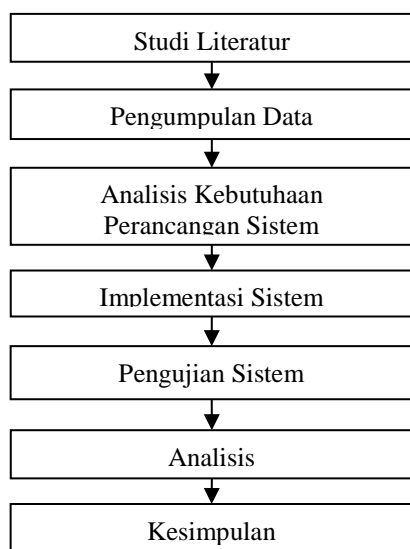
Algoritme Genetika adalah algoritme optimasi yang terinspirasi oleh seleksi alami dan genetika alami. Dalam Algoritme Genetika dalam pencarian solusi yang optimal, proses

pencarian dilakukan di antara sejumlah alternatif titik optimal berdasarkan fungsi probabilistik (Mahmudy, 2008). Populasi awal dibangkitkan secara acak sedangkan populasi berikutnya berasal dari hasil evaluasi dan seleksi dari kromosom yang terbaik. Pada proses pemilihan kromosom terbaik pada setiap generasi digunakan alat ukur yaitu *fitness*. Setiap generasi akan menghasilkan kromosom-kromosom baru yang dibentuk dari generasi sebelumnya dengan menggunakan operator reproduksi (*reproduction*), kawin silang (*crossover*), dan juga mutasi (*mutation*). Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*) (Adipranata, et al., 2007). Dalam algoritme terdapat 3 parameter yang harus didefinisikan yaitu ukuran populasi (*popsiz*), *Mutation Rate* (*Mr*) dan *Crossover Rate* (*Cr*).

#### 4. METODOLOGI PENELITIAN

##### 4.1. Tahapan Penelitian

Dalam penelitian ini dilakukan dengan beberapa tahapan-tahapan yang diilustrasikan pada gambar blok diagram yang dijelaskan pada Gambar 1.



Gambar 1. Blok diagram metodologi penelitian

##### 4.2. Data yang Digunakan

Data-data yang digunakan dalam penelitian ini meliputi data mata pelajaran, data guru, data kelas, data jadwal penggunaan ruang lab, data jam pelajaran dan data SK Penugasan guru mengajar pada tahun ajaran 2016-2017.

##### 4.3. Alur Penyelesaian Masalah Menggunakan Algoritme Genetika

Dari data penugasan guru mengajar harus dilakukan *encoding* terlebih dahulu, *encoding* perlu dilakukan untuk mengubah bentuk data asli yang masih mentah menjadi data yang dapat diproses oleh algoritme genetika. *Encoding* data dilakukan untuk membuat data dengan kode yang unik yang mewakili seluruh data penugasan. Dari data *encoding* dengan kode yang unik tersebut selanjutnya dapat dilakukan proses pembentukan kromosom sebagai elemen dalam algoritme genetika. Selain data-data penjadwalan, sistem juga memerlukan masukan berupa parameter-parameter algoritme genetika yaitu jumlah populasi, iterasi maksimal atau jumlah generasi, nilai *Crossover Rate* (*Cr*) dan nilai *Mutation Rate* (*Mr*).

##### 4.4. Representasi Kromosom

Gen-gen yang akan dibangun dalam representasi kromosom berupa bilangan *integer* yang merupakan kode unik dari data penugasan guru mengajar sehingga akan membentuk sebuah kromosom yang disusun secara *random*. Panjang dari kromosom adalah sepanjang banyaknya data dalam penugasan guru mengajar. Contoh representasi kromosom dijelaskan pada Gambar 2.

Kromosom	Indeks	1	2	3	4	5	6	7
	Gen	12	6	8	9	14	2	1
	Indeks	8	9	10	11	12	13	14
	Gen	4	11	13	10	3	7	5

Gambar 2. Contoh Representasi Kromosom

##### 4.5. Perhitungan Nilai *Fitness*

Nilai dari *fitness* ditentukan berdasarkan jumlah pelanggaran yang terjadi dari hasil pembentukan jadwal yang dihasilkan oleh setiap kromosom. Rumus dalam perhitungan nilai *fitness* menggunakan Persamaan (1) dan Persamaan (2).

$$f(x) = \sum_{i=1}^n (\text{Penalti}_i * \text{Bobot}_i) \quad (1)$$

Keterangan:

$\text{Penalti}_i$  = Jumlah pelanggaran terhadap *constraint* ke-*i*

$\text{Bobot}_i$  = Bobot pelanggaran *constraint* ke-*i*

$$\text{fitness} = \frac{1}{1+f(x)} \quad (2)$$

Keterangan:

$f(x)$  = Fungsi obyektif (Total nilai bobot seluruh pelanggaran).

Untuk menentukan nilai pelanggaran didasarkan pada jenis-jenis pelanggaran yang terjadi. Jenis-jenis pelanggaran dijelaskan pada Tabel 4.

**Tabel 4.** Jenis Pelanggaran

Jenis	Pelanggaran	Bobot
<i>Hard Constraint</i>	Seorang guru tidak boleh berada pada ruang dan jam yang sama pada satu waktu	0.0035
	Satu ruang lab hanya digunakan oleh satu kelas saja dalam satu waktu	0.0035
<i>Soft Constraint</i>	Mata pelajaran olahraga harus berada antara jam ke 1 hingga jam ke 6	0.00075
	Seorang guru yang mengajar pada 2 lokasi Jl. Veteran dan Jl. Monginsidi harus selisih minimal 1 jam pelajaran	0.00075
	Dalam satu hari yang sama terdapat maksimal 4 jam untuk satu mata pelajaran yang sama	0.00075
	Guru yang melakukan <i>request</i> untuk hari dan jam yang tidak bisa tidak boleh dijadwalkan pada hari dan jam tersebut	0.00075

#### 4.6. Crossover

*Crossover* dilakukan dengan cara menyilangkan 2 induk untuk menghasilkan keturunan. Dalam proses persilangannya, metode yang digunakan adalah *one-cut-point crossover* yaitu dengan cara mengambil beberapa gen dari induk pertama yang kemudian digabungkan dengan gen pada induk kedua. Jumlah *child* dari proses reproduksi ditentukan dengan menentukan nilai *Cr* (*Crossover Rate*) yang dibangun dari nilai antara 0 hingga 1. Jumlah *offspring* pada proses *crossover* dicari menggunakan Persamaan (3).

$$offspringcross = Ceil(Cr * popsize) \quad (3)$$

Keterangan:

*Cr* = *Crossover Rate*

*Popsiz* = Jumlah individu dalam setiap generasi

*Ceil* = Fungsi untuk pembulatan

#### 4.7. Mutasi

Mutasi merupakan proses reproduksi dengan cara memodifikasi susunan gen dari suatu induk. Dari satu proses mutasi terhadap satu induk akan menghasilkan satu *child*. Sebelum dilakukan proses reproduksi menggunakan mutasi perlu menentukan nilai *Mr* (*Mutation Rate*) terlebih dahulu yang dibangkitkan dari nilai antara 0 dan 1 yang mana fungsi dari *Mr* ini adalah untuk menentukan berapa jumlah *child* yang harus dihasilkan pada proses mutasi. Jumlah *offspring* pada proses mutasi dicari menggunakan Persamaan (4).

$$offspringmut = Ceil(Mr * popsize) \quad (4)$$

Keterangan:

*Cr* = *Crossover Rate*

*Popsiz* = Jumlah individu dalam setiap generasi

*Ceil* = Fungsi untuk pembulatan

#### 4.8. Evaluasi

Proses evaluasi dilakukan penggabungan seluruh individu dalam generasi dan seluruh *child* dari hasil reproduksi. Dalam proses evaluasi juga dilakukan penghitungan nilai *fitness* dari seluruh kumpulan individu dan *child* yang kemudian dilakukan evaluasi terhadap nilai *fitness* dari seluruh kumpulan individu. *Output* dari fungsi *fitness* ini akan digunakan sebagai dasar untuk menentukan individu yang masuk pada generasi selanjutnya (Syamsuddin, 2004).

#### 4.9. Seleksi

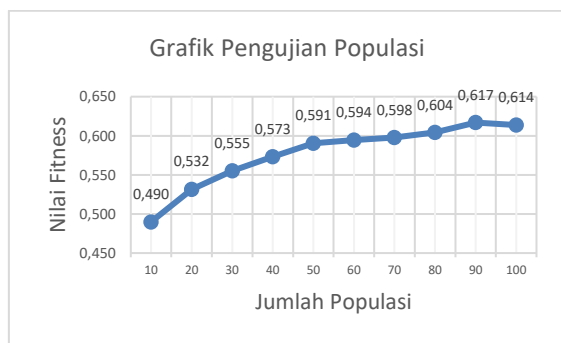
Tahap seleksi dilakukan proses pemilihan individu yang akan masuk ke generasi selanjutnya. Proses pemilihan individu dilakukan dengan metode *elitism selection*. Metode *elitism selection* dilakukan dengan cara memilih individu-individu dengan nilai *fitness* tertinggi. Dari kumpulan individu tersebut diambil sebanyak *popsiz* dengan nilai *fitness* paling tinggi yang kemudian menjadi populasi baru untuk menjadi generasi selanjutnya.

### 5. PENGUJIAN DAN ANALISIS

#### 5.1. Pengujian Ukuran Populasi

Pengujian ukuran populasi dilakukan dengan cara melakukan uji coba terhadap nilai

10 sampai 100 dengan rentang masing-masing adalah 10. Setiap percobaan terhadap satu nilai populasi dilakukan uji coba sebanyak 10 kali hal ini dilakukan karena hasil dari setiap uji coba bisa menghasilkan nilai *fitness* yang berbeda-beda yang disebabkan oleh pembangkitan populasi awal secara acak atau *random*. Grafik hasil uji coba pada Gambar 3.

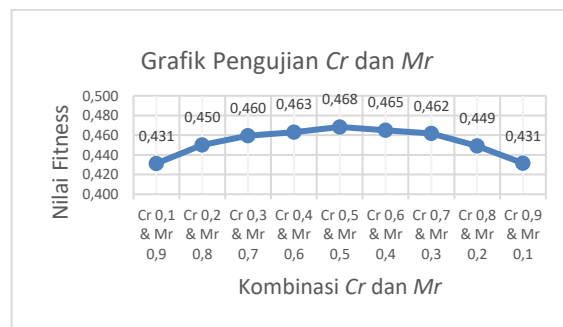


**Gambar 3.** Grafik Hasil Pengujian Ukuran Populasi

Gambar 3 menunjukkan bahwa meningkatkan ukuran populasi mulai dari 10 sampai 50 sangat mempengaruhi peningkatan nilai *fitness* yang dihasilkan secara sangat signifikan. Ketika dilakukan percobaan dengan meningkatkan ukuran populasi menjadi 60 sampai 80 dihasilkan bahwa nilai *fitness* yang dihasilkan terjadi peningkatan yang sangat kecil dan tidak begitu signifikan. kembali terdapat peningkatan yang begitu signifikan ketika ukuran populasi pada 90 akan tetapi ketika ukuran populasi pada angka 1000 terjadi penurunan nilai *fitness* yang dihasilkan. Dari percobaan tersebut didapatkan hasil bahwa jika populasi maksimum yang didapatkan adalah 90 dan jika ditambahkan lagi ukuran populasi yang lebih besar tidak memiliki kemungkinan besar untuk mendapatkan nilai *fitness* yang semakin besar pula.

## 5.2. Pengujian Nilai *Cr* dan *Mr*

Proses pengujian dilakukan dengan cara membuat kombinasi nilai *Crossover Rate* (*Cr*) dan *Mutation Rate* (*Mr*) mulai dari 0,1 sampai 0,9. Setiap kombinasi nilai *Crossover Rate* (*Cr*) dan *Mutation Rate* (*Mr*) kemudian dilakukan percobaan sebanyak 10 kali. Grafik hasil uji coba ditunjukkan pada Gambar 4.



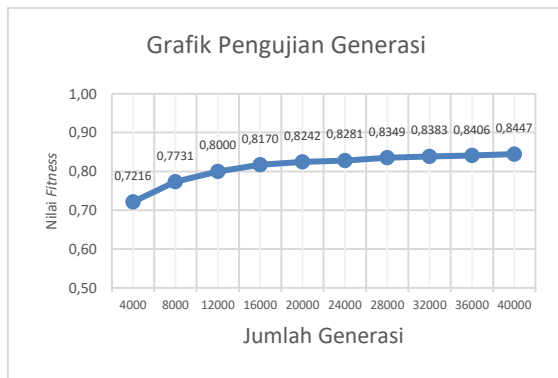
**Gambar 4.** Grafik Hasil Pengujian Nilai *Cr* dan *Mr*

Pada Gambar 4 menunjukkan bahwa setiap perubahan pada nilai *Cr* dan *Mr* juga sangat mempengaruhi nilai *fitness* yang dihasilkan. Dari percobaan yang telah dilakukan menunjukkan bahwa terjadi peningkatan nilai *fitness* pada nilai *Cr* 0,1 dan *Mr* 0,9 sampai pada nilai *Cr* 0,5 dan *Mr* 0,5. Akan tetapi setelah itu mengalami penurunan nilai *fitness* pada nilai *Cr* 0,5 dan *Mr* 0,5 sampai pada nilai *Cr* 0,9 dan *Mr* 0,1. Hal tersebut menunjukkan bahwa untuk menghasilkan nilai *fitness* yang optimal diperlukan kombinasi dari nilai *Cr* dan *Mr* yang seimbang. Jika hanya nilai *Cr* yang tinggi maka dalam proses pencarian solusinya cenderung bersifat eksploitasi yaitu setiap proses reproduksi hanya dihasilkan anak yang mirip dengan induknya yang dapat mengakibatkan keragaman individu semakin berkurang. Sedangkan jika hanya nilai *Mr* yang tinggi maka dalam proses pencarian solusinya bersifat eksplorasi dan dapat menghasilkan anak yang unik dari individu sebelumnya.

## 5.3. Pengujian Jumlah Generasi

Pengujian Jumlah Generasi digunakan untuk menentukan jumlah generasi yang paling sesuai untuk dapat menghasilkan nilai *fitness* paling optimal. Pengujian ini dilakukan dengan cara melakukan uji coba sebanyak 10 kali untuk masing-masing nilai jumlah generasi yang diuji coba kan yaitu antara 4.000 sampai 40.000 dengan masing-masing percobaan rentang nilainya adalah 4.000. Grafik hasil uji coba ditunjukkan pada Gambar 5.





**Gambar 5.** Grafik Hasil Pengujian Jumlah Generasi

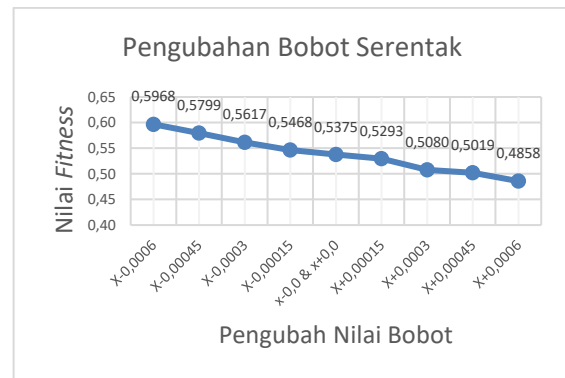
Pada Gambar 5 menunjukkan bahwa ukuran jumlah generasi sangat mempengaruhi nilai dari *fitness* yang dihasilkan. Semakin banyak ukuran dari generasi maka semakin luas juga kesempatan untuk melakukan pencarian solusi optimal (Mahmudy, 2013). Pada saat jumlah generasi di antara 4000 dan 28000 masih terjadi peningkatan nilai *fitness* yang signifikan, akan tetapi pada saat jumlah generasi berada pada di antara 28000 sampai 40000 masih terjadi peningkatan nilai *fitness* tetapi tidak begitu signifikan dan cenderung meningkat secara stabil hal tersebut dapat dikarenakan bahwa pada iterasi 28000 sudah mulai terjadi konvergensi. Dari percobaan dengan jumlah total generasi 40000 terjadi terus mengalami peningkatan nilai *fitness* walaupun peningkatan yang terjadi semakin sedikit. Hal tersebut dikarenakan pada generasi sudah mengalami keadaan yang konvergen sehingga kemungkinan untuk dihasilkan individu yang lebih baik juga semakin kecil.

#### 5.4. Pengujian Pembobotan *Constraint*

Beberapa skenario pengujian yang dilakukan untuk pengujian pembobotan *constraint* antara lain:

##### 5.4.1. Pengujian Mengubah Nilai Bobot Serentak

Pengujian mengubah nilai bobot serentak dilakukan dengan cara mengubah nilai bobot serentak dilakukan dengan cara mengubah nilai bobot baik untuk *hard constraint* maupun *soft constraint* secara bersamaan dan dengan nilai pengubah yang sama. Proses pengubahan bobot dilakukan dengan cara menaikkan dan menurunkan secara berkala dengan selisih nilai setiap perubahan adalah 0,00015. Grafik hasil uji coba ditunjukkan pada Gambar 6.

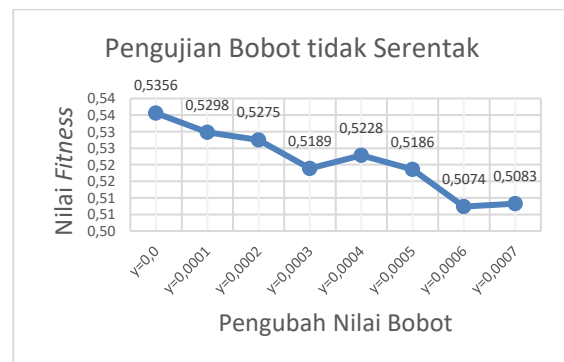


**Gambar 6.** Grafik Hasil Pengujian Pembobotan Serentak

Berdasarkan grafik pada Gambar 6 menunjukkan bahwa pada saat nilai bobot tidak ditambahkan maupun dikurangkan maka nilai *fitness* berada ditengah-tengah dari *range* nilai *axis* yaitu 0,537. Akan tetapi jika nilai dari bobot untuk seluruh *constraint* dilakukan pengurangan maka grafik akan naik secara signifikan dan akan terjadi kondisi yang sebaliknya jika nilai bobot dari seluruh *constraint* dilakukan penambahan maka nilai *fitness* yang dihasilkan akan menurun walaupun dari grafik total pelanggaran terdapat perubahan jumlah pelanggaran baik untuk *hard constraint* maupun *soft constraint*.

##### 5.4.2. Pengujian Mengubah Nilai Bobot Tidak Serentak

Pengujian mengubah nilai bobot tidak serentak dilakukan dengan cara mengubah nilai bobot dengan cara menaikkan untuk bobot *hard constraint* dan menurunkan nilai bobot untuk *soft constraint*. Nilai pengurang maupun penambah yang digunakan yaitu nilai dengan kelipatan 0,0001. Grafik hasil uji coba ditunjukkan pada Gambar 7.



**Gambar 7.** Grafik Hasil Pengujian Pembobotan Tidak Serentak

Berdasarkan grafik pada Gambar 7 menunjukkan bahwa jika pada pengujian

pembobotan tidak serentak walaupun dilakukan penambahan nilai bobot untuk *hard constraint* akan tetapi terdapat pengurangan bobot untuk *soft constraint* dan tetap dihasilkan nilai fitness yang menurun. Berdasarkan kondisi tersebut dapat ditarik kesimpulan bahwa jumlah pelanggaran pada *hard constraint* sangat lebih dominan dari pada jumlah pelanggaran yang terjadi pada *soft constraint*.

## 6. KESIMPULAN

Beberapa tahapan yang digunakan dalam proses implementasi Algoritme Genetika dalam pencarian solusi jadwal mata pelajaran yaitu proses inisialisasi kromosom, Kromosom dapat dibangkitkan dari data penugasan guru dengan representasi bilangan *integer* yang setiap gennya berisi kode-kode penugasan yang telah dibangkitkan secara random. Setiap kromosom dengan nilai *fitness* tertinggi merupakan representasi dari solusi jadwal mata pelajaran. proses reproduksi menggunakan *one-cut point crossover* dan *reciprocal exchange mutation*, proses evaluasi dan perhitungan *fitness* dan yang terakhir adalah proses seleksi menggunakan *elitism selection*.

Berdasarkan dari hasil pengujian yang telah dilakukan didapatkan beberapa nilai parameter-parameter Algoritme Genetika yaitu nilai jumlah populasi terbaik adalah 90, nilai kombinasi *Cr* dan *Mr* adalah 0.5 dan 0.5, dan jumlah generasi sebanyak 40000. Proses pencarian solusi dengan menggunakan parameter-parameter tersebut didapatkan nilai *fitness* yaitu sebesar 0,8451

## DAFTAR PUSTAKA

- Adipranata, R., Soedjianto, F. & Tjondro, W., 2007. Perbandingan Algoritma Exhaustive, Algoritma Genetika Dan Algoritma Jaringan Syaraf Tiruan Hopfield Untuk Pencarian Rute Terpendek.
- Herjanto, E., 2006. *Manajemen Operasi*. 3rd penyunt. Jakarta: Grasindo.
- Jain, A., Jain, S. & Chande, 2010. Formulation of Genetic Algorithm to Generate Good Quality Course Timetable. *IACSIT*, Volume 248-251, p. 1.
- Mahmudy, W. F., 2008. Optimasi fungsi tanpa kendala menggunakan algoritma genetika dengan kromosom biner dan perbaikan

chromosome hill-climbing. *Kursor*, Volume 5, pp. 23-29.

- Mahmudy, W. F., 2013. *Algortima Genetika*. Malang: Program Teknologi Informasi Daln Ilmu Komputer (PTIIK) Universitas Brawijaya Malang.
- Mulyana, A., 2010. *Manajemen Kurikulum SMP Terbuka di Kecamatan Tenjo-Bogor*. S1: UIN Syarif Hidayatullah Jakarta.
- Promcharoen, S., Pana, W. & Tanhan, P., 2009. Genetic Algorithm for Exam. *ICROS-SICE International Joint Conference*, pp. 2903-2906.
- Syamsuddin, A., 2004. *Pengenalan Algoritma Genetika*. s.l.:Kuliah Umum IlmuKomputer.Com.
- Wen, S. & Wen-jun, S., 2006. Decision of Production Planning in Agricultural Management. pp. 645-650.