

PERFORMA OPTIMAL PENERAPAN ALGORITMA GENETIKA PADA PENJADWALAN MATA KULIAH

Heri Hermawan^{1*)}, Ahmad Fauzi¹⁾, Yana Cahyana¹⁾, Hanny Hikmayanti Handayani¹⁾

¹⁾Teknik Informatika, Universitas Buana Perjuangan Karawang, Karawang

*Email Korespondensi : if16.herihermawan@mhs.ubpkarawang.ac.id

ABSTRAK

Penjadwalan mata kuliah sangat diperlukan dalam perkuliahan, terdapat beberapa kendala dalam proses penjadwalan mata kuliah di universitas. Ketersediaan ruangan dan slot waktu perlu diperhitungkan karena hal tersebut merupakan faktor yang dapat mempersulit proses penjadwalan dan dapat terjadi bentrok dalam jadwal. Pada penelitian ini, algoritma genetika digunakan untuk menyelesaikan permasalahan optimasi penjadwalan mata kuliah. Data yang digunakan adalah data perkuliahan seperti data mata kuliah, data dosen, data ruangan dan waktu. Berdasarkan hasil pengujian yang dilakukan terhadap nilai parameter, jumlah generasi dan nilai *fitness* algoritma genetika yaitu 157 individu, jumlah generasi 11, probabilitas *crossover* 0,1, dan probabilitas mutasi 0,3 menghasilkan penjadwalan mata kuliah yang optimal dengan nilai *fitness* terbaik 1 tanpa terjadi bentrok. Hasil penjadwalan mata kuliah yang optimal tersebut tidak terlepas dari optimalnya parameter yang digunakan.

Kata kunci: Algoritma Genetika, Mata Kuliah, Nilai Fitness, Optimasi, Penjadwalan

ABSTRACT

Course scheduling is very necessary for lectures, there are several obstacles in the process of scheduling courses at universities. The availability of space and time slots need to be taken into account because these are factors that can complicate the scheduling process and may clash in schedules. In this study, a genetic algorithm was used to solve the optimization problem of course scheduling. The data used are lecture data such as course data, lecturer data, room data, and time. Based on the results of tests carried out on the parameter value, the number of generations and the fitness value of the genetic algorithm, namely 157 individuals, the number of generations 11, the crossover probability of 0.1, and the mutation probability of 0.3 resulted in optimal course scheduling with the best fitness value of 1 without any clashes. The optimal course scheduling results cannot be separated from the optimal parameters used.

Keywords: Genetic Algorithms, Courses, Fitness Scores, Optimization, Scheduling

PENDAHULUAN

Penjadwalan merupakan sebuah pengaturan waktu dalam suatu kegiatan operasi, seperti mengalokasikan fasilitas, peralatan atau tenaga kerja, dan menentukan urutan pelaksanaan kegiatan operasi [1]. Begitu pun dengan penjadwalan mata kuliah di universitas harus terjadwal dengan baik agar proses belajar mengajar tidak terhambat oleh penjadwalan yang tidak tersusun dengan baik. Penjadwalan yang optimal merupakan penjadwalan di mana seluruh pihak dalam proses belajar dan mengajar dapat dilakukan. Bukan hanya dosen yang mengajar, juga mahasiswa yang mengampu mata kuliah tersebut [2]. Banyak kendala yang dihadapi ketika proses penjadwalan mata kuliah di universitas. Banyaknya mahasiswa dan sedikitnya ruangan yang tersedia menjadikan proses penjadwalan menjadi sulit. Hal tersebut perlu diperhitungkan dengan baik, agar tidak mempersulit proses penjadwalan dan terciptanya penjadwalan yang baik dan tepat. Banyak faktor yang mempengaruhi proses penjadwalan mata kuliah seperti faktor dosen, ruang kelas, dan jam yang tersedia. Jadwal yang telah disusun saat ini masih terjadi bentrok ruang

kelas, antar dosen mata kuliah maupun antar kelas [1]. Ditambah proses penjadwalan mata kuliah harus dilakukan setiap pergantian semester. Salah satu cara untuk mengatasi masalah tersebut adalah dengan menerapkan optimasi penjadwalan mata kuliah menggunakan teknik komputasi.

Penyelesaian optimasi penjadwalan sudah dilakukan oleh peneliti sebelumnya menggunakan algoritma genetika. Algoritma genetika memberikan solusi terbaik dan merupakan sebuah teknik optimasi dalam ilmu komputer. Algoritma genetika menggunakan teknik yang terinspirasi dari Biologi Evolusioner dan mempertimbangkan kondisi tahapan seperti warisan, mutasi (*mutation*), seleksi alam (*selection*), dan rekombinasi (*crossover*) [3]. Penelitian sebelumnya telah dilakukan oleh Desiana [1], yaitu mengenai penjadwalan mata pelajaran dengan menggunakan algoritma genetika. Parameter yang digunakan yaitu sebanyak 100 jumlah generasi, 50 individu, 0.7 nilai probabilitas *crossover* dan 0.1 nilai probabilitas mutasi. Hasil pengujian yang dilakukan dari kinerja algoritma genetika tersebut cukup baik dengan menghasilkan kinerja pada generasi ke-80 ditemukan hanya 2 total *crash* / bentrok dengan nilai *fitness* terbaik 0,19452381. Penelitian terdahulu lainnya yang telah dilakukan oleh Robbi dan Nurochman [4], algoritma genetika berhasil diimplementasikan untuk penjadwalan instruktur training ICT dengan parameter jumlah populasi 30 individu, probabilitas *crossover* (*Pc*) 0.4 dan 0.1 nilai probabilitas mutasi (*Pm*). Pada penelitian tersebut menghasilkan nilai *fitness* 0.9523 dengan 1 nilai *error* pada *constraint* pembagian ruang kelas dengan bobot 0.05. Selanjutnya penelitian yang dilakukan oleh Sugeha, Inkiriwang dan Pingkan [5], yaitu mengenai sistem penjadwalan proyek menggunakan algoritma genetika yang berhasil diterapkan dan dapat menjadi alternatif keputusan bagi kontraktor dalam pelaksanaan proyek. Penelitian tersebut menghasilkan penjadwalan dengan waktu yang lebih singkat dari jadwal yang sebenarnya yaitu diselesaikan selama 3 minggu, dan di singkat menjadi 2 minggu masa pekerjaan. Kemudian penelitian yang dilakukan oleh Rohaeni, Saepudin, Rohmawati, dan Plengan [6], yaitu algoritma genetika mengenai optimasi debit air pada pembangkit listrik tenaga air. Penelitian tersebut menghasilkan, pada tahun 2013 *outflow* yang telah dioptimasi dapat menghasilkan listrik sebesar 13,619,115 kw dan kekurangan 0 m³. Berdasarkan *outflow* dari data aktual dapat menghasilkan listrik sebesar 13,619,115 kw dan kekurangan 9,201,883 m³. *Outflow* hasil optimasi dengan menggunakan algoritma genetika lebih baik dari *outflow* data aktual.

Kemudian penelitian mengenai perbandingan performa algoritma genetika dan ant colony optimization dalam penjadwalan kuliah yang dilakukan oleh Kurniati, Rahmatulloh and Rahmawati [7]. Algoritma ant colony optimization lebih sedikit memakan memori dibandingkan algoritma genetika yaitu 33% dan lebih sedikit pula waktu yang dibutuhkan dibandingkan dengan algoritma genetika yaitu 82% lebih cepat. Namun nilai Fitness yang di hasilkan algoritma genetika lebih baik dibandingkan dengan algoritma ant colony optimization yaitu 11% lebih baik, artinya dalam penjadwalan kuliah algoritma genetika lebih baik dan sedikit bentrok dibandingkan algoritma ant colony optimization.

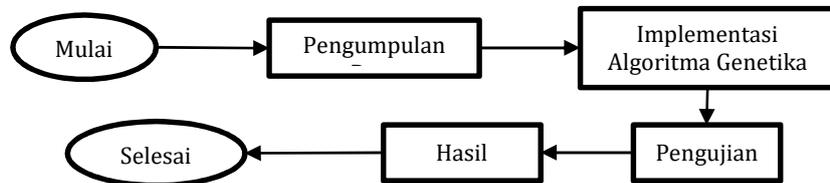
Berdasarkan permasalahan yang telah dipaparkan dan beberapa penelitian yang telah dibuktikan, maka dilakukan penelitian menggunakan algoritma genetika untuk mendapatkan performa yang optimal mengenai penerapan algoritma genetika pada penjadwalan mata kuliah.

METODE PENELITIAN

Gambaran Umum Penelitian

Metode penelitian merupakan gambaran sebuah langkah atau cara yang akan ditempuh untuk melakukan penelitian. Pada penelitian ini akan menerapkan algoritma

genetika untuk menyelesaikan permasalahan penjadwalan mata kuliah di universitas. Gambar 1 merupakan gambaran penelitian yang akan dilaksanakan.



Gambar 1. Gambaran Umum Penelitian

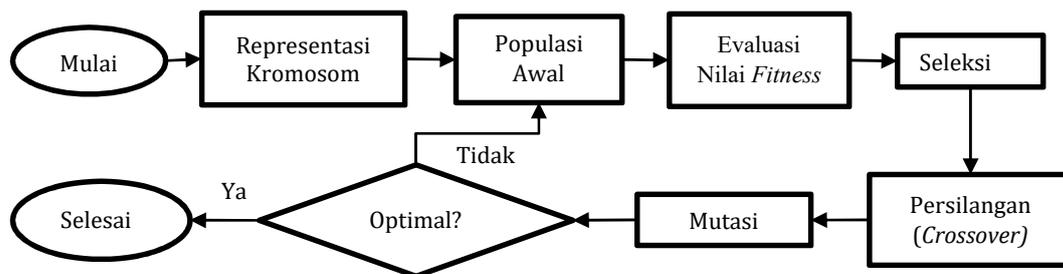
Pengumpulan Data

Data yang digunakan dalam proses penjadwalan mata kuliah di Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan Karawang menggunakan algoritma genetika yaitu data penjadwalan mata kuliah Program Studi Teknik Informatika tahun akademik 2019/2020 semester ganjil. Data tersebut kemudian disusun berdasarkan jenisnya agar mempermudah dalam proses penjadwalan mata kuliah. Berikut data – data yang sudah dikelompokkan berdasarkan jenisnya:

- a. Data Kuliah = 157
- b. Data Ruangan = 14 ruangan yang tersedia dalam perkuliahan 1 hari
- c. Data slot waktu yang tersedia dari senin – sabtu = 41 slot

Algoritma Genetika

Algoritma genetika merupakan teknik pencarian (*searching technique*) dan teknik optimasi yang cara kerjanya meniru proses evolusi dan perubahan struktur genetik pada makhluk hidup dengan prinsip dasar cara kerja algoritma ini diilhami oleh proses seleksi alam dan prinsip - prinsip dari ilmu genetika [5]. Penyelesaian berbagai masalah optimasi dari masalah sederhana hingga rumit pun dapat diselesaikan dengan baik menggunakan algoritma genetika [1]. Kemampuan algoritma genetika untuk menemukan solusi yang bagus secara cepat terhadap masalah yang berkarakteristik seperti ruang masalah sangat besar, kompleks, dan sulit dipahami [8]. Tahapan algoritma genetika ditunjukkan pada Gambar 2.



Gambar 2. Tahapan Algoritma Genetika

Representasi Kromosom

Representasi kromosom merupakan pengkodean kromosom, di mana data atau bahan penjadwalan dikodekan agar mempermudah dalam proses pembentukan populasi awal dan tahap algoritma genetika selanjutnya. Pengkodean kromosom dalam penelitian ini menggunakan pengkodean bilangan riil (*Permutation Encoding*). Pengkodean dilakukan dengan mengkodekan setiap data yang nantinya akan dijadikan setiap gen dalam kromosom. Pengkodean yang dilakukan pada penelitian ini adalah pengkodean kuliah, pengkodean ruangan dan pengkodean waktu. Tahap pengkodean data kuliah ditunjukkan pada Tabel 1.

Tabel 1. Kode Data Kuliah

Kode	Mata Kuliah	Kelas	Dosen
1	Komputer & Masyarakat	IF16A	Euis Nurlaelasari, M.Kom
2	Komputer & Masyarakat	IF16B	Euis Nurlaelasari, M.Kom
3	Komputer & Masyarakat	IF16C	Euis Nurlaelasari, M.Kom
...
155	Pengantar Teknologi Informasi	IF19D	Kiki Ahmad B, M.Kom
156	Pengantar Teknologi Informasi	IF19E	Kiki Ahmad B, M.Kom
157	Pengantar Teknologi Informasi	IF19F	Kiki Ahmad B, M.Kom

Pengkodean ruangan dilakukan dengan mengetahui jumlah dari ruangan yang akan dikodekan. Total ruangan pada Tabel 2 merupakan total ruangan yang digunakan dalam proses perkuliahan selama 1 hari. Pengkodean ruangan dilakukan dari bilangan 1 hingga total ruangan seperti pada Tabel 2.

Tabel 1 Kode Ruangan

Kode	Ruangan
1	A403
2	A301
3	A303
4	A302
5	A304
...	...
12	D407
13	B408
14	A404

Pengkodean waktu dilakukan dengan menggabungkan data hari dari hari senin hingga sabtu dengan data waktu dalam satu hari dari jam 08.00 hingga 20.00. Penggabungan dilakukan dengan mengkali data hari = 6 hari × data waktu data dalam satu hari = 7 waktu = 42 slot waktu. Data tersebut dipotong 1 slot waktu untuk beribadah shalat Jum'at. Demikian slot waktu yang tersedia adalah 41 slot waktu. Kemudian mengkodekan total slot waktu dengan bilangan 1 hingga total slot waktu seperti pada Tabel 3:

Tabel 2. Kode Waktu

Kode	Hari	Waktu
1	Senin	08.00
2	Senin	10.00
3	Senin	12.00
...
39	Sabtu	16.00
40	Sabtu	18.00
41	Sabtu	20.00

Populasi Awal

Kromosom dibangkitkan sebanyak jumlah kuliah. Pembangkitan individu untuk masing masing kromosom yang terdiri dari gen kuliah, gen ruang, dan gen waktu.

Tabel 3. Populasi Awal

	Individu 1	individu 2	individu 3	individu 4	...	individu 157
Kromosom 1	0,0,38	1,6,26	2,12,23	3,2,39	...	156,4,37
Kromosom 2	0,12,0	1,2,6	2,6,29	3,4,26	...	156,9,23
Kromosom 3	0,5,36	1,13,1	2,6,7	3,9,28	...	156,12,0
Kromosom 4	0,12,9	1,9,19	2,10,13	3,13,17	...	156,6,19

Setiap kromosom merupakan gabungan dari individu, dan panjang kromosom merupakan sebanyak jumlah kuliah. Kuliah merupakan data penggabungan antara mata kuliah, kelas dan dosen, seperti pada Tabel 1. Setiap individu terdiri dari 3 komponen yang terdiri dari kuliah, ruangan dan waktu. Pada kromosom 1 individu 1 nilai 0 pada baris pertama merupakan komponen kuliah, 0 kedua merupakan komponen ruangan, 38 merupakan komponen waktu, dan begitu pun susunan seterusnya hingga individu terakhir. Individu [0,0,38] pada kromosom 1 memiliki arti kuliah ke 1, ruangan ke 1 dan waktu ke 37. Adapun jumlah kromosom tergantung inputan. Semakin banyak kromosom, maka semakin banyak pilihan jadwal. Namun, jumlah kromosom juga harus diperhatikan karena, semakin banyak jumlah kromosom, maka akan semakin lambat dalam proses pembuatan jadwal.

Evaluasi Nilai *Fitness*

Perhitungan *fitness* didasarkan pada jumlah *penalty* atau pelanggaran yang telah ditentukan sebelumnya pada setiap individu dalam kromosom.

Tabel 4. Aturan Penalty

Constrain	Keterangan	Nilai Penalty
1.	Dosen yang sama diwaktu yang sama.	1
2.	Kelas yang sama diwaktu yang sama	1
3.	Ruang yang sama diwaktu yang sama	1

Proses perhitungan *fitness* dilakukan dengan cara menghitung semua jumlah penalti yang terjadi disetiap kromosom dengan nilai penalti 1 untuk melanggar dan 0 jika tidak melanggar. Rumus aturan perhitungan *fitness* ditunjukkan pada persamaan (1).

$$\text{Nilai fitness} = \frac{1}{(1 + (F_1 B_1 + \dots + F_n B_n))} \quad (1)$$

Keterangan:

- F_n = Banyaknya pelanggaran ke-n.
- B_n = Bobot pelanggaran ke-n.

Seleksi

Metode seleksi yang digunakan pada penelitian ini adalah seleksi roda rolet (*roulette wheel selection*). Seleksi *roulette wheel selection* induk dipilih berdasarkan nilai *fitness* dari kromosom itu sendiri. Lebih baik suatu kromosom maka besar kesempatan terpilih. Melalui seleksi kita dapat memilih jadwal yang terbaik dari jadwal yang tersedia. Tahap seleksi pertama adalah menghitung probabilitas dari setiap *fitness* kromosom. Nilai probabilitas didapatkan dari nilai *fitness* setiap kromosom yang dibagi dengan total nilai *fitness* seluruh kromosom. Rumus seleksi yang digunakan ditunjukkan pada persamaan (2).

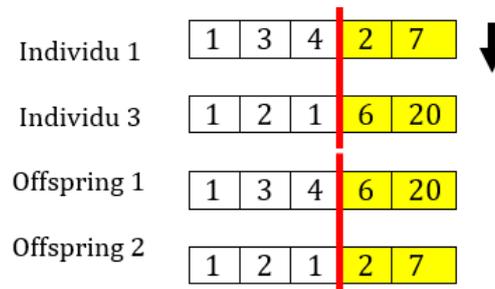
$$P(i) = \frac{f_i}{N \sum_{j=1}^{f_j}} \quad (2)$$

Keterangan:

- $P(i)$ = Probabilitas fitness ke-i
- f_i = Nilai fitness individu ke-i.

Persilangan (*Crossover*)

Metode persilangan (*Crossover*) yang digunakan dalam penelitian ini adalah *one point crossover*. Pada metode *one point crossover* tidak semua induk akan mengalami proses *crossover* karena pemilihan induk melibatkan parameter probabilitas *crossover* (P_c). Misalkan kromosom 3 dan kromosom 4 yang akan di *crossover* dengan *crossover rate* 50%. *Crossover* kromosom 3 berada di titik pemotongan ke 78. Pemotongan titik kromosom 3 diambil dari 1 – 78 dari kromosom 3 dan dari 79 – 157 diambil dari kromosom 4. Begitu pun sebaliknya dengan *crossover* kromosom ke 4. Ilustrasi *crossover* ditunjukkan pada Gambar 3.



Gambar 3. Ilustrasi *One Point Crossover*

Mutasi

Penggantian gen atau mutasi terhadap gen dalam kromosom yang digantikan hanya ruang dan waktunya saja, untuk kuliuhnya tetap. Jumlah gen yang akan diganti tergantung *mutation rate*-nya. Terdapat 4 kromosom dengan panjang individu 157. jumlah individu tersebut yang akan dimutasi berdasarkan jumlah *mutation rate*. Misalkan, *mutation rate* 25% maka, jumlah individu yang akan di mutasi adalah: Total individu = 4×157 (jumlah kromosom \times jumlah individu per kromosom) = 628 individu. kemudian jumlah yang akan dimutasi adalah: Jumlah Mutasi = $25\% \times 628$ (mutation rate \times jumlah individu) = 157 individu. Proses mutasi dilakukan dengan membangkitkan bilangan acak antara 1 - 628 (total individu) sebanyak 157 kali. Apabila bilangan acak pertama 200 maka, kromosom ke 2 individu ke 43 (karena kromosom pertama sampai 157 dan sisa 43 diambil dari kromosom ke 2). Sehingga ruang dan waktu pada individu 43 kromosom ke 2 akan diganti dengan mengambil ruang dan waktu secara acak.

Pengujian

Pengujian algoritma genetika dilakukan untuk menghasilkan performa optimal dan nilai *fitness* yang mendekati optimum. Terdapat beberapa pengujian yang akan dilakukan yaitu pengujian nilai parameter, pengujian jumlah generasi dan pengujian nilai *fitness*. Pengujian dilakukan setelah semua iterasi selesai dilakukan dan akan dihitung nilai *fitness*-nya.

HASIL DAN PEMBAHASAN

Pengujian dilakukan terhadap aplikasi penjadwalan mata kuliah algoritma genetika yang telah dibuat. Data yang digunakan merupakan data penjadwalan yaitu data kuliah seperti pada Tabel 1, data ruangan seperti pada Tabel 2, dan data waktu seperti pada Tabel 3. Terdapat beberapa pengujian yang akan dilakukan untuk mendapatkan nilai *fitness* mendekati optimum.

Pengujian nilai parameter

Pengujian parameter dilakukan untuk menemukan parameter yang terbaik. Nilai parameter tersebut berpengaruh untuk mendapatkan nilai *fitness* terbaik. Nilai parameter tersebut yaitu nilai parameter probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m). Percobaan dilakukan pada nilai probabilitas *crossover* (P_c) dari 0,1 hingga 0,9 dengan probabilitas mutasi (P_m) dari 0,1 hingga 0,9 dengan banyak peluang percobaan hingga 81 dari perpaduan probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m). Percobaan tersebut dilakukan sebanyak 10 kali dengan 10 generasi, 5 kromosom dan jumlah individu sebanyak data kuliah seperti pada Tabel 6. Probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m) yang terbaik ditunjukkan dengan nilai rata - rata probabilitas tertinggi. Hasil pengujian parameter ditunjukkan pada Tabel 6.

Tabel 6. Hasil Pengujian Nilai Parameter

Pc	Pm	Generasi	Nilai Fitness						Rata
			1	2	3	4	...	10	
0,8	0,2	10	0,11111	0,03125	0,2	0,02941	...	0,05	0,105441
0,1	0,3	10	0,05882	0,5	0,05882	0,2	...	0,0625	0,151822
0,2	0,4	10	0,05556	0,07143	0,1	0,03704	...	0,33333	0,105109
0,9	0,4	10	0,03704	0,125	0,08333	0,04348	...	0,03846	0,116541
...
0,4	0,5	10	0,08333	0,05	0,06667	0,25	...	0,16667	0,104091
0,2	0,6	10	0,05	0,05	0,08333	0,07143	...	0,09091	0,109259
0,1	0,7	10	0,125	0,05882	0,07143	0,125	...	0,25	0,102717
0,2	0,9	10	0,33333	0,16667	0,06667	0,09091	...	0,05263	0,133382

Hasil pengujian nilai parameter pada penelitian ini menghasilkan nilai probabilitas terbaik yaitu probabilitas *crossover* (Pc) 0,1 dan probabilitas mutasi (Pm) 0,3 dengan rata-rata 0,151822. Hasil tersebut menunjukkan bahwa nilai probabilitas *crossover* (Pc) 0,1 dan probabilitas mutasi (Pm) 0,3 merupakan probabilitas terbaik.

Pengujian jumlah generasi dan nilai *fitness*

Pengujian dilakukan terhadap 100 generasi, 0,1 probabilitas *crossover* (Pc) dan probabilitas mutasi (Pm) 0,3 yang didapat dari hasil pengujian nilai parameter dengan jumlah individu sebanyak data kuliah seperti pada Tabel 7.

Tabel 7. Parameter Pengujian Jumlah Generasi Dan Nilai *Fitness*

Nama Parameter	Nilai Parameter
Jumlah Kromosom	5
Generasi	100
Probabilitas Crossover (pc)	0,1
Probabilitas Mutasi (pc)	0,3

Mula-mula jumlah individu dibangkitkan sebanyak data kuliah yaitu 157 dan banyak kromosom 5 dengan parameter pengujian seperti pada Tabel 7. Pengujian dilakukan sebanyak 10 kali terhadap total generasi yaitu 100. Pengujian bertujuan untuk mendapatkan nilai *fitness* mendekati optimum dari jumlah individu yang diuji. Adapun hasil dari data pengujian disajikan pada Tabel 8.

Tabel 8. Hasil Pengujian Nilai *Fitness* Terbaik

Pengujian Ke-	Generasi (solusi ditemukan)	Total Generasi	Total Crash (Bentrok)	Nilai <i>Fitness</i> terbaik
1	13	100	0	1
2	11	100	0	1
3	14	100	0	1
4	13	100	0	1
5	15	100	0	1
6	14	100	0	1
7	17	100	0	1
8	12	100	0	1
9	15	100	0	1
10	15	100	0	1

Berdasarkan Tabel 8 hasil dari pengujian pada 100 generasi, 157 individu, probabilitas *crossover* (Pc) 0,1 dan probabilitas mutasi (Pm) 0,3, diperoleh nilai *fitness* terbaik pada pengujian ke 2 di generasi ke 11 adalah 1, hal tersebut menunjukkan bahwa tidak terjadi bentrok (0 *crash*) pada hasil jadwal.

KESIMPULAN

Performa yang dihasilkan algoritma genetika penjadwalan mata kuliah dengan 157 individu, parameter probabilitas *crossover* 0,1 dan probabilitas mutasi 0,3 dengan jumlah generasi maksimal 100. Menghasilkan nilai fitness terbaik yaitu 1 pada generasi terendah yaitu ke 11. Nilai *fitness* 1 menunjukkan bahwa tidak terjadi bentrok (crash) pada hasil penjadwalan yang dibuat. Keberhasilan penerapan algoritma genetika mendapatkan nilai fitness terbaik tidak terlepas dari penerapan parameter dan jumlah generasi yang dipilih.

Saran untuk penelitian selanjutnya yaitu, penentuan jumlah individu yang digunakan harus sebanding dengan jumlah kelas dan slot waktu yang tersedia. Hal tersebut yang menjadi kemungkinan terjadinya penjadwalan tanpa bentrok (crash). Penerapan algoritma genetika penjadwalan mata kuliah pada penelitian ini tidak mempertimbangkan kesediaan waktu dosen untuk mengajar. Pembangkitan solusinya semua dosen dalam keadaan bersedia untuk mengajar tanpa menanyakan waktu kesediaan mengajar.

UCAPAN TERIMA KASIH

Naskah ilmiah ini adalah sebagian dari penelitian Tugas Akhir milik Heri Hermawan yang dibimbing oleh Ahmad Fauzi dan Yana Cahyana.

REFERENSI

- [1] E. Desiana, "Performance Algoritma Genetika (GA) Pada Penjadwalan Mata Pelajaran," *InfoTekJarJurnal Nas. Inform. dan Teknol. Jaringan*. e-ISSN 2540-7600, pp. 56-60, 2016.
- [2] L. A. Aditya and W. Mega PD, "Implementasi algoritma genetika untuk penjadwalan mata pelajaran pada lms getsmart," *Junal Manaj. dan Inform. Pelita Nusantara*. ISSN2088-3943, vol. 21, no. 1, pp. 65-70, 2017.
- [3] N. H. Hari, F. Prasetyo, and Hamdlani, "Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika di Sekolah Menengah Kejuruan Annuqayah - Sumenep," *J. Sist. Inf.* . ISSN 2579-5341, vol. 5341, no. October, pp. 66-74, 2018.
- [4] N. Robbi and Nurochman, "Implementasi Algoritma Genetika Untuk Penjadwalan Instruktur Training ICT UIN Sunan Kalijaga," *JISKa*. ISSN 2527 -5836, vol. 1, no. 3, pp. 123-132, 2017.
- [5] I. Sugeha, R. L.Inkiriwang, and P. Pingkan, "Optimasi penjadwalan menggunakan metode algoritma genetika pada proyek rehabilitasi puskesmas minanga," *J. Sipil Statik*. ISSN 2337-6732, vol. 7, no. 12, 2019.
- [6] L. Rohaeni, D. Saepudin, A. A. Rohmawati, and P. Plengan, "Penerapan Algoritma Genetika Untuk Optimasi Debit Air Pada Pembangkit Listrik Tenaga Air Studi Kasus : Waduk Situ Cileunca , Jawa Barat," *e-Proceeding Eng*. ISSN 2355-9365, vol. 3, no. 2, pp. 3961-3971, 2016.
- [7] N. I. Kurniati, A. Rahmatulloh, and D. Rahmawati, "Perbandingan Performa Algoritma Koloni Semut Dengan Algoritma Genetika - Tabu Search Dalam Penjadwalan," *J. Comput. Eng. Syst. Sci*. p-ISSN 2502-7131, vol. 4, no. 1, pp. 17-23, 2019.
- [8] M. Munirah and Subanar, "Kajian terhadap Beberapa Metode Optimasi (Survey of Optimization Methods)," *JUITA* p-ISSN: 2086-9398; e-ISSN: 2579-8901, vol. V, pp. 45-50, 2017.