

# Penjadwalan Mata Kuliah Menggunakan Metode *Hybrid* Algoritma Genetika Dan Algoritma Koloni Semut

Muhammad Arief Priambodo <sup>1</sup>, Fhira Nhita <sup>2</sup>, Annisa Aditsania <sup>3</sup>

School of Computing, Telkom University  
Jl. Telekomunikasi No. 01, Bandung, Indonesia

<sup>1</sup> riphh.cool@gmail.com

<sup>2</sup> fhiranhita@telkomuniversity.ac.id

<sup>3</sup> annisaaditsania@gmail.com

## Abstract

*Lectures timetabling has its own difficulties because there are many things need to be considered. These considerations include number of students, number of class available, number of lecturers who are not equal to the number of subjects and lectures which has been determined. In this study, lectures timetabling at the Telkom University in Bandung was implemented using the hybridization method with algorithm genetic adaptive and ant colony optimization. Ant colony optimization integrated into the genetic algorithm that has been built in advance to determine the probability of crossover and mutation probability values in which the two procedures are part of the genetic algorithm. Based on hybridization algorithm testing that has been done, the value of the solution reached 88.24%, which shows the number of scheduling solution. With these test results, indicating that the hybridization method with adaptive genetic algorithm and ant colony optimization can minimize problem in lectures scheduling.*

**Keywords:** scheduling, genetic algorithm, ant colony algorithm.

## Abstrak

Pengaturan penjadwalan mata kuliah memiliki tantangan tersendiri dikarenakan banyaknya pertimbangan-pertimbangan yang harus diperhatikan. Pertimbangan tersebut diantaranya adalah jumlah mahasiswa, jumlah ruangan, jumlah dosen yang tidak sebanding dengan jumlah mata kuliah, serta waktu perkuliahan yang telah ditentukan. Dalam penelitian kali ini, penjadwalan mata kuliah di Universitas Telkom Bandung diimplementasikan menggunakan metode hibridisasi antara algoritma genetika adaptif dengan algoritma koloni semut. Algoritma koloni semut dipadukan kedalam algoritma genetika yang sudah dibangun terlebih dahulu untuk menentukan nilai probabilitas crossover dan nilai probabilitas mutasi yang dimana dua prosedur tersebut merupakan bagian dari algoritma genetika. Berdasarkan pengujian algoritma hibridisasi yang telah dilakukan, diperoleh nilai solusi tercapai sebesar 88.24%, dimana solusi tercapai menunjukkan jumlah penjadwalan yang tidak mengalami bentrok. Dengan hasil pengujian tersebut, menunjukkan bahwa dengan metode hibridisasi algoritma genetika adaptif dengan algoritma koloni semut dapat meminimalkan bentrokan yang terjadi pada penjadwalan.

**Kata Kunci:** penjadwalan, algoritma genetika, algoritma koloni semut.

## I. PENDAHULUAN

Pentingnya suatu jadwal dalam sebuah kegiatan tidak lain agar kegiatan tersebut berjalan dengan lancar tanpa adanya gangguan seperti bentroknnya jadwal. Penyusunan jadwal memiliki tantangan tersendiri, karena penjadwalan harus dibuat sedemikian rupa sehingga tidak ada lebih dari satu kegiatan pada waktu yang bersamaan.

Penjadwalan mata kuliah pada suatu Institusi/Unviversitas bukanlah perkara yang dapat dikatakan mudah karena banyak sekali faktor yang dapat mempengaruhi jadwal tersebut. Diantaranya adalah jumlah mahasiswa, jumlah ruangan, jumlah dosen yang tidak sebanding dengan jumlah mata kuliah, serta waktu perkuliahan yang telah ditentukan. Dalam kasus kali ini, permasalahan optimasi penjadawalan diselesaikan dengan menggunakan algoritma

genetika dan algoritma koloni semut. Algoritma koloni semut dalam kasus kali ini bertugas untuk menentukan nilai probabilitas *crossover* ( $P_c$ ) dan nilai probabilitas mutasi ( $P_m$ ) yang dimana dua prosedur tersebut merupakan bagian dari algoritma genetika yang berguna untuk membentuk kromosom baru yang harapannya kromosom tersebut lebih baik dari pada induknya.

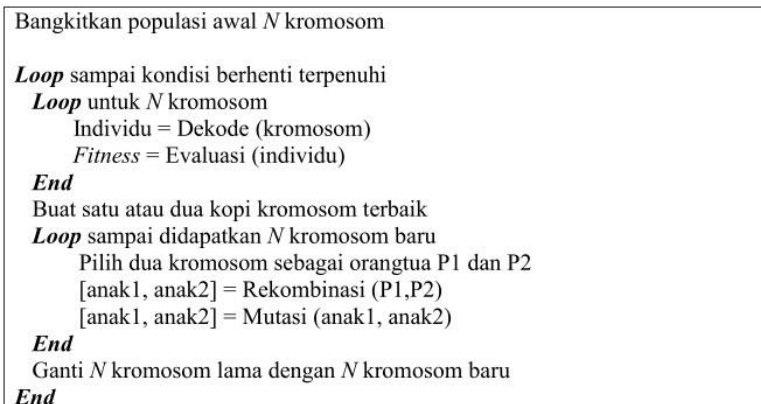
Penelitian ini ditujukan untuk memberikan solusi penjadwalan dengan meminimalkan bentroknya penjadwalan yang dikerjakan pada mata kuliah tingkat satu diseluruh Fakultas Teknik Universitas Telkom Bandung tahun ajaran 2014/2015.

## II. METODE PENELITIAN

Terdapat 2 metode yang digunakan pada penelitian kali ini, yaitu algoritma genetika dan algoritma koloni semut. Algoritma koloni semut digunakan untuk menentukan nilai probabilitas *crossover* ( $P_c$ ) dan nilai probabilitas mutasi ( $P_m$ ) yang dimana dua prosedur tersebut merupakan bagian dari algoritma genetika yang berguna untuk membentuk kromosom baru.

### A. Algoritma Genetika

Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan pada tahun 1975 di Amerika Serikat [1]. Konsep dasar yang mengilhami timbulnya algoritma genetika adalah teori evolusi alam yang dikemukakan oleh Charles Darwin. Dalam teori tersebut dijelaskan bahwa pada proses evolusi alami, setiap individu harus melakukan adaptasi terhadap lingkungan disekitarnya agar dapat bertahan hidup [1]. Secara umum skema algoritma genetika ditunjukkan pada Gambar 1.



Gambar 1. Algoritma Genetika

1) *Representasi individu*: Tahap pertama pada algoritma genetika yaitu mengetahui tipe data seperti apa yang akan kita teliti untuk selanjutnya diproses kedalam skema pengkodean. Yang dimana skema pengkodean ini nantinya akan merepresentasikan setiap kromosom yang akan kita teliti. Terdapat beberapa representasi kromosom diantaranya adalah representasi biner, representasi bilangan bulat *integer*, representasi real (*floating point*), dan representasi permutasi [2].

2) *Nilai fitness*: Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai kualitasnya. Penghitungan dilakukan dengan memberikan pinalti untuk setiap aturan yang digunakan dalam penjadwalan [1].

$$F = \frac{1}{1 + \sum N_p N_b}$$

(1)

$N_p$  adalah jumlah pelanggaran dan  $N_b$  adalah nilai bobot pada setiap pelanggaran yang terjadi.

3) *Seleksi Orang Tua*: Pemilihan kromosom sebagai calon orang tua, dilakukan berdasarkan nilai *fitness* dari setiap kromosom. Kromosom yang memiliki nilai *fitness* yang besar, memiliki peluang terpilihnya sebagai orang tua lebih besar pula [2]. Walaupun probabilitas kromosom yang lainnya lebih kecil tetapi tetap memiliki kemungkinan terpilih sebagai orang tua.

4) *Rekombinasi*: Setelah kromosom–kromosom terpilih sebagai orang tua maka dilakukanlah rekombinasi yang sering disebut *crossover*, yang bertujuan untuk menghasilkan kromosom baru/anak (*offspring*) yang dimana diharapkan memiliki nilai *fitness* yang lebih baik daripada kromosom sebelumnya/orang tua (*parent*). Tidak semua kromosom (*parent*) melalui proses ini karena *crossover* sendiri memiliki probabilitas yang telah ditentukan sebelumnya dan dinyatakan dengan variable *Pc*, tetapi dalam penelitian kali ini nilai dari *Pc* akan ditentukan melalui algoritma koloni semut.

5) *Mutasi*: Setelah melakukan proses *crossover* selanjutnya dilakukan proses mutasi, dimana mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom [2]. Seperti halnya pada proses *crossover*, proses mutasi pun memiliki probabilitas yang dinyatakan dengan *Pm*, yang akan ditentukan pula melalui algoritma koloni semut.

6) *Seleksi Survivor*: Setelah mendapatkan kromosom–kromosom yang dihasilkan melalui proses *crossover* dan proses mutasi, maka semua kromosom lama akan digantikan dengan kromosom yang baru tanpa memperhatikan nilai *fitness*-nya proses ini dinamakan seleksi *general replacement*.

#### B. Algoritma Koloni Semut

Mengacu pada kasus yang akan diteliti dimana algoritma koloni semut ini akan digunakan untuk mendapatkan nilai *Pc* dan *Pm* yang terdapat di dalam algoritma genetika maka cara kerja algoritma koloni semut yang digunakan, mengacu pada jurnal [3]. Adapun langkah utama dari algoritma yang diusulkan diberikan pada *pseudo code* dibawah ini.

```

Inisialisasi
Mengisi harga parameter :
  Kriteria berhenti : jumlah iterasi (T)
  Membangkitkan bilangan acak (X); //solusi dari fitness
  Menentukan fitness;
  Mencari nilai  $f_{initial}^{best}$  dan  $X_{initial}^{best}$ ;
  Inisialisasi feromon trail ( $\tau$ ); //mengikuti persamaan 2-4
  Menentukan nilai  $\alpha_{initial}$ ;

Solusi looping
For j=1 to T
  For i=1 to m // jumlah semut
    Bangkitkan bilangan acak  $d_x$  dengan interval  $[-\alpha, \alpha]$ ;
    Hitung nilai X yang baru; //mengikuti persamaan 5
    Hitung nilai fitness;
  end
  if  $f(X_T^{best}) \leq f(X_{T-1}^{best})$  then
     $X^{global} = X_T^{best}$ ;
  else
     $X^{global} = X_{T-1}^{best}$ ;
  Update nilai feromon ( $\tau$ ); //mengikuti persamaan 2-4
   $\alpha_T = (rand(0.1, 0.9) \times \alpha_{T-1})$  //update nilai  $\alpha$ 

end

```

Gambar 2. *Pseudo code* AKS

Mengacu kepada jurnal [3], berikut adalah tahapan untuk mendapatkan nilai feromon yang sudah dijelaskan pada Gambar 2, terlebih dahulu mencari nilai variabel  $\phi$  dan  $\phi$ .

$$f_i = f_i - f_{min}$$

(2)

$f_i$  merupakan selisih antara  $f_i$  dan  $f_{min}$ . Dimana  $f_i$  merupakan *fitness* ke- $i$  dan  $f_{min}$  adalah *fitness minimum*. Setelah mendapatkan nilai  $f_i$  berikutnya akan dicari nilai  $\phi_i$  yang dijelaskan pada persamaan 3.

$$\phi_i = \frac{D_i}{2 + f_i}$$

(3)

Setelah mendapatkan nilai  $\phi_i$  dengan nilai  $t$  sebesar 0.005 [3], yang dimana variabel  $\phi_i$  akan digunakan untuk menghitung nilai feromon, yang ditunjukkan pada persamaan 4.

$$\tau_i = \frac{\phi_i}{\sum_{i=1}^n \phi_i}$$

(4)

$\tau_i$  adalah nilai dari feromon ke- $i$ . Feromon sendiri digunakan sebagai tolak ukur untuk mendapatkan  $X_k$  dan  $X_{k-1}$ . Pada persamaan 4 berikut adalah cara untuk mendapatkan nilai  $X$ , yang merupakan solusi dari *fitness*. Berikut adalah untuk mendapatkan nilai  $X$  yang baru [3].

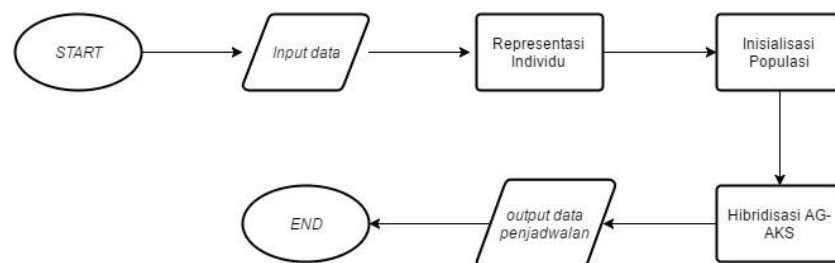
$$X_k = X_{k-1} + \tau_i$$

(4)

$X_k$  adalah solusi sebelumnya berdasarkan nilai feromon, variabel  $k$  menunjukan semut ke- $k$ , dan variabel  $t-1$  menunjukan iterasi sebelumnya. Nilai  $\tau_i$  berupa bilangan acak yang berada pada rentang  $[-u, u]$ .

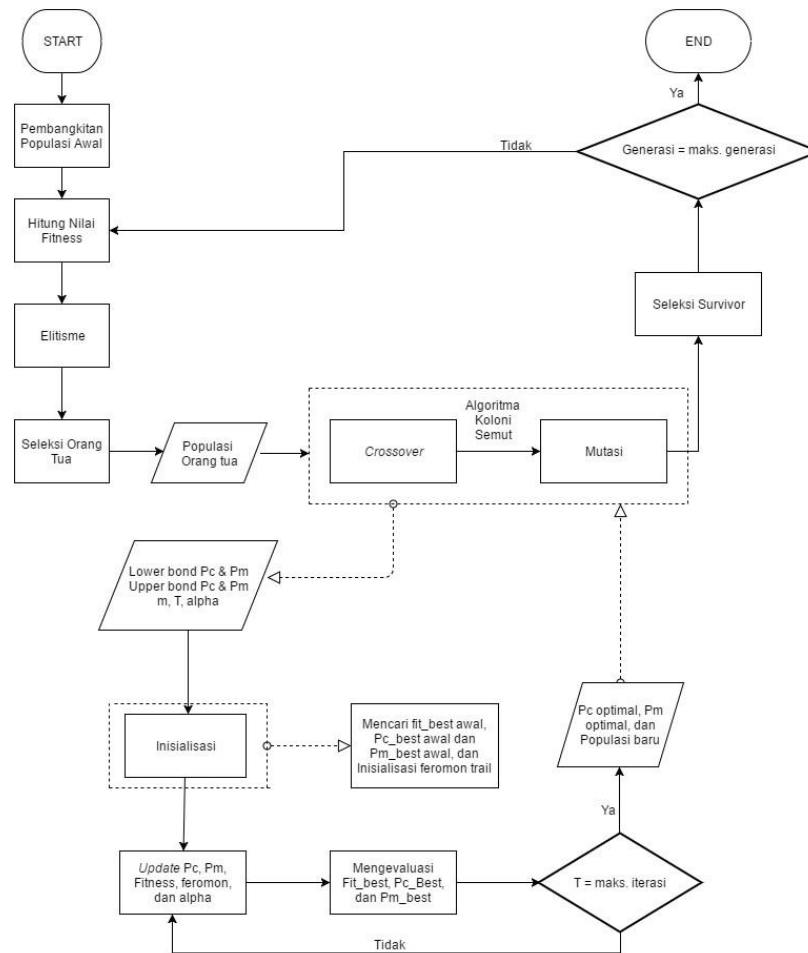
### III. Desain sistem

Dalam penelitian ini, dilakukan perancangan sistem untuk membuat penjadwalan mata kuliah dengan menggunakan algoritma genetika dan algoritma koloni semut. Pencarian solusi untuk masalah penjadwalan ini menggunakan data acuan yang diberikan oleh pihak PPDU, yaitu data mata kuliah untuk seluruh Fakultas Teknik Universitas Telkom. Dalam sistem ini terdapat beberapa proses, pertama-tama data terlebih dahulu dilakukan *pre-processing*, berikutnya data yang sudah dilakukan *pre-processing* akan dijadikan *input*-an untuk algoritma genetika yang didalam prosesnya terdapat algoritma koloni semut yang akan bertugas untuk melakukan proses *crossover* dan proses mutasi, yang dimana  $P_c$  dan  $P_m$  juga ditentukan oleh algoritma koloni semut yang berarti nilai dari  $P_c$  dan  $P_m$  akan bergerak dinamis seiring iterasi yang ditentukan didalam algoritma koloni semut itu sendiri. Adapun skema perancangan sistem secara umum adalah sebagai berikut.



Gambar 3. Skema umum

Secara umum metode algoritma genetika dijalankan terlebih dahulu, selanjutnya disisipkan algoritma koloni semut guna untuk melakukan proses *crossover* dan proses mutasi, yang dimana  $P_c$  dan  $P_m$  juga ditentukan oleh algoritma koloni semut tersebut.



Gambar 4. AG-AKS

Gambar diatas merupakan *flowchart* dari metode algoritma genetika yang dihybrid dengan algoritma koloni semut. Ketika algoritma genetika sudah melalui proses seleksi orang tua dan terbentuk populasi orang tua, maka populasi tersebut akan menjadi sebuah *input*-an bagi algoritma koloni semut. Selanjutnya algoritma koloni semut akan mencari sebuah nilai  $P_c$  dan  $P_m$  untuk nantinya akan dilakukan proses *crossover* dan proses mutasi. Kemudian sebelum proses kembali lagi kedalam proses algoritma genetika, algoritma koloni semut melakukan evaluasi terlebih dahulu terhadap nilai *fitness* yang baru dan terhadap nilai  $P_c$  dan  $P_m$  yang didapat.

#### IV. Hasil Pengujian dan diskusi

Pengujian dalam sistem ini bertujuan untuk menganalisa perbedaan hasil solusi yang dikeluarkan oleh metode-metode yang dimiliki, yaitu metode hibridisasi AG-AKS, dan algoritma genetika yang dimana nantinya terlihat perbandingan terhadap jumlah bentrokan yang terjadi dalam penjadwalan perkuliahan.

##### A. Skenario 1

Pertama-tama pada skenario 1 kali ini akan dilakukan pengujian terhadap metode AG sebanyak empat kali, dengan kombinasi parameter  $UkPop$  dan  $MaxGen$  sejumlah 600 populasi. Nilai dari parameter  $P_c$  dan  $P_m$  yang digunakan pada metode AG masing-masing sebesar 0.6 dan 0.01

Tabel 1. Skenario 1 AG

| No. | Populasi | Maks. Generasi | Solusi Tercapai | <i>Fitness</i> |
|-----|----------|----------------|-----------------|----------------|
| 1.  | 15       | 40             | 72.83 %         | 0.002582       |
| 2.  | 40       | 15             | 74.23 %         | 0.002668       |
| 3.  | 20       | 30             | 73.95 %         | 0.002642       |
| 4.  | 30       | 20             | 75.07 %         | 0.002727       |

Hasil nilai rata-rata solusi tercapai terbesar dari Tabel 1 adalah pada pengujian ke-4, yaitu sebesar 75.07% dengan nilai *fitness* 0.002727. Sedangkan untuk hasil nilai rata-rata solusi tercapai terendah ditunjukkan pada pengujian ke-1, yaitu sebesar 72.83% dengan nilai *fitness* 0.002582.

Berikutnya dilakukan pengujian terhadap metode AGA-AKS sebanyak 12 kali. Pada pengujian ini kombinasi parameter *UkPop* dan *MaxGen* dengan jumlah sebesar 600 populasi akan dipadukan dengan ukuran parameter T dan m sejumlah 100 semut buatan. Masing-masing nilai *Pc* dan *Pm* pada metode AG-AKS ada pada rentang [0.6-0.9] dan [0.01-0.2].

Tabel 2. Skenario AG-AKS

| No. | Populasi | Maks. Generasi | T  | m  | Solusi Tercapai | <i>Fitness</i> |
|-----|----------|----------------|----|----|-----------------|----------------|
| 1.  | 40       | 15             | 20 | 5  | 85.43%          | 0.0016         |
| 2.  | 40       | 15             | 5  | 20 | 78.15%          | 0.0013         |
| 3.  | 40       | 15             | 10 | 10 | 83.57%          | 0.0017         |
| 4.  | 15       | 40             | 20 | 5  | 84.59%          | 0.0018         |
| 5.  | 15       | 40             | 5  | 20 | 81.23%          | 0.0016         |
| 6.  | 15       | 40             | 10 | 10 | 84.31%          | 0.0019         |
| 7.  | 20       | 30             | 20 | 5  | 87.67%          | 0.0021         |
| 8.  | 20       | 30             | 5  | 20 | 86.27%          | 0.0021         |
| 9.  | 20       | 30             | 10 | 10 | 86.27%          | 0.0021         |
| 10. | 30       | 20             | 20 | 5  | 87.67%          | 0.0021         |
| 11. | 30       | 20             | 5  | 20 | 77.31%          | 0.0013         |
| 12. | 30       | 20             | 10 | 10 | 79,83%          | 0.0015         |

Hasil nilai rata-rata solusi tercapai terbesar dari Tabel 2 adalah pada pengujian ke-5, yaitu sebesar 88.24% dengan nilai *fitness* 0.0023. Sedangkan untuk hasil nilai rata-rata solusi tercapai terendah ditunjukkan pada pengujian ke-11, yaitu sebesar 77.31% dengan nilai *fitness* 0.0013.

Berdasarkan pengujian skenario 1, yang ditunjukkan pada Tabel 1 dan Tabel 2. Bahwa solusi tercapai terbesar ada pada Tabel 4.2 yang ditunjukkan oleh pengujian ke-5 yang dimana pengujian tersebut dilakukan dengan metode

hibridisasi AG-AKS. Dengan begitu dapat terlihat bahwa metode hibridisasi AG-AKS memiliki solusi lebih optimal dibandingkan dengan metode algoritma genetika.

### B. Skenario 2

Pada skenario 2, pengujian terhadap kedua metode dilakukan dengan mencatat nilai dari *fitness* rata-rata dan *fitness* maksimum pada setiap generasinya. Pada penelitian kali ini *MaxGen* yang ditentukan sebelumnya adalah sebesar 10 generasi dengan jumlah *UkPop* sebesar 40 populasi. Pada algoritma genetika untuk nilai dari *Pc* dan *Pm* masing-masing sebesar 0.6 dan 0.01. Pada Tabel 3 menunjukkan hasil skenario 2 pada metode algoritma genetika.

Tabel 3. Hasil AG

| Generasi ke- | <i>Pc</i> | <i>Pm</i> | <i>fitness</i><br>rata-rata | <i>fitness</i> |
|--------------|-----------|-----------|-----------------------------|----------------|
| 1.           | 0.6       | 0.01      | 0.000758                    | 0.001161       |
| 2.           | 0.6       | 0.01      | 0.000766                    | 0.001161       |
| 3.           | 0.6       | 0.01      | 0.000777                    | 0.001161       |
| 4.           | 0.6       | 0.01      | 0.000802                    | 0.001161       |
| 5.           | 0.6       | 0.01      | 0.000852                    | 0.001161       |
| 6.           | 0.6       | 0.01      | 0.000862                    | 0.001161       |
| 7.           | 0.6       | 0.01      | 0.000912                    | 0.001161       |
| 8.           | 0.6       | 0.01      | 0.000914                    | 0.001161       |
| 9.           | 0.6       | 0.01      | 0.000927                    | 0.001328       |
| 10.          | 0.6       | 0.01      | 0.000938                    | 0.001328       |

Nilai dari parameter *UkPop* dan *MaxGen* pada metode hibridisasi AG-AKS sama dengan metode algoritma genetika. Untuk jumlah *m* dan *T* masing-masing sebesar 10. Pada Tabel 4 menunjukkan hasil yang diperoleh skenario 2 dari metode AG-AKS.

Tabel 4. Hasil AG-AKS

| Generasi ke- | $P_c$  | $P_m$  | $fitness$<br>rata-rata | $fitness$ |
|--------------|--------|--------|------------------------|-----------|
| 1.           | 0.9000 | 0.0100 | 0.000759               | 0.001161  |
| 2.           | 0.6563 | 0.2000 | 0.000764               | 0.001161  |
| 3.           | 0.9000 | 0.0100 | 0.000775               | 0.001261  |
| 4.           | 0.9000 | 0.2000 | 0.000783               | 0.001261  |
| 5.           | 0.9000 | 0.0890 | 0.000833               | 0.001261  |
| 6.           | 0.7655 | 0.0710 | 0.000858               | 0.001372  |
| 7.           | 0.8219 | 0.0100 | 0.000863               | 0.001372  |
| 8.           | 0.8899 | 0.2000 | 0.000893               | 0.001372  |
| 9.           | 0.9000 | 0.0331 | 0.000921               | 0.001372  |
| 10.          | 0.9000 | 0.1751 | 0.000940               | 0.001372  |

Berdasarkan pengujian skenario 2, pada Tabel 4.3 *Fitness* maks terbesar sebesar 0.001328 dan Tabel 4.4 *Fitness* maks terbesar sebesar 0.001372. Dengan begitu dapat terlihat bahwa metode hibridisasi AG-AKS memiliki *Fitness* maks lebih besar dibandingkan dengan metode algoritma genetika.

### C. Skenario 3

Pada pengujian skenario 3 dilakukan pengujian sebanyak 16 kali dengan setiap pengujian dilakukan sebanyak tiga kali dan diambil nilai rata-rata dari hasil setiap pengujian yang dilakukan. Pengujian dilakukan terhadap nilai parameter  $P_c$  dan  $P_m$  agar mengetahui pengaruhnya terhadap persentase solusi tercapai dan nilai *fitness* yang dihasilkan pada metode algoritma genetika. Karena pada algoritma genetika nilai parameter  $P_c$  dan  $P_m$  tidak bergerak dinamis seperti halnya pada metode algoritma AG-AKS. Pada algoritma genetika terlebih dahulu harus dimasukkan satu persatu nilai dari parameter  $P_c$  dan  $P_m$ . Masing-masing nilai  $P_c$  dan  $P_m$  ada pada rentang [0.6-0.9] dan [0.01-0.2]. Dengan parameter  $UkPop$  sebanyak 20 dan  $MaxGen$  20. Pada Tabel 5 menunjukkan hasil pengujian yang dilakukan.

Tabel 5. Skenario 3

| Generasi ke- | $P_c$ | $P_m$ | $fitness$ | Solusi Tercapai |
|--------------|-------|-------|-----------|-----------------|
| 1.           | 0.6   | 0.01  | 0.001033  | 71.71%          |
| 2.           | 0.7   | 0.01  | 0.001067  | 71.71%          |
| 3.           | 0.8   | 0.01  | 0.001233  | 75.35%          |
| 4.           | 0.9   | 0.01  | 0.001100  | 72.55%          |
| 5.           | 0.6   | 0.05  | 0.001100  | 71.99%          |
| 6.           | 0.7   | 0.05  | 0.001133  | 73.95%          |
| 7.           | 0.8   | 0.05  | 0.001233  | 75.35%          |
| 8.           | 0.9   | 0.05  | 0.001233  | 75.07%          |
| 9.           | 0.6   | 0.10  | 0.001333  | 77.31%          |
| 10.          | 0.7   | 0.10  | 0.001233  | 73.95%          |
| 11.          | 0.8   | 0.10  | 0.001233  | 73.67%          |
| 12.          | 0.9   | 0.10  | 0.001233  | 74.79%          |
| 13.          | 0.6   | 0.20  | 0.001195  | 76.47%          |
| 14.          | 0.7   | 0.20  | 0.001100  | 71.43%          |
| 15.          | 0.8   | 0.20  | 0.001200  | 73.11%          |
| 16.          | 0.9   | 0.20  | 0.001100  | 70.59%          |

Pada Tabel 4.5, hasil yang optimal ditunjukkan pada pengujian ke-9 dengan ukuran parameter  $P_c$  sebesar 0.6 dan  $P_m$  sebesar 0.10, yang memiliki nilai rata-rata solusi tercapai sebesar 77.31% dan nilai rata-rata  $fitness$  sebesar 0.001333. Sedangkan untuk hasil yang terendah ditunjukkan pada pengujian ke-1 dengan nilai rata-rata solusi tercapai sebesar 71.71% dan nilai rata-rata  $fitness$  0.001033.

## V. Kesimpulan

Setelah dilakukan beberapa skenario pengujian, dapat disimpulkan bahwa metode hibridisasi AG-AKS mampu menambahkan tingkat performansi didalam penjadwalan matakuliah. Terlihat dari hasil setiap solusi tercapai yang dihasilkan lebih tinggi dibanding algoritma genetika.

**REFERENSI**

- [1] Witary, Vinny., Rachmat., Nur, dan Inayatullah. Optimasi Penjadwalan Perkuliahan dengan Menggunakan Algoritma Genetika (Studi Kasus : AMIK MDP, STMIK GI MDP dan STIE MDP).
- [2] Suyanto. 2008. Evolutionary Computation, Komputasi Berbasis “Evolusi” dan “Genetika”. Informatika, Bandung.
- [3]Seçkiner, S.U., dkk. 2013. “Ant colony optimization for continuous functions by using novel pheromone updating” . Department of Industrial Engineering, Faculty of Engineering, University of Gaziantep, Turki.